IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

BALLY MANUFACTURING CORPORATION,

Plaintiff,

v.

D. GOTTLIEB & CO., a corporation
WILLIAMS ELECTRONICS, INC., a corporation, and
ROCKWELL INTERNATIONAL CORPORATION, a corporation

Defendants.

CIVIL ACTION NO.

78 C 2246

AFFIDAVIT OF ZANE A. SANDUSKY IN SUPPORT OF ROCKWELL'S MOTION TO DISMISS UNDER RULE 12, FED. R. CIV. P.

Zane A. Sandusky, being duly swirn, deposes and says that:

(1) Since 1955 he has been an engineer with Rockwell International Corporation (hereafter Rockwell). Since prior to 1970 he has been associated with electronic controllers for various types of systems, and since 1976 he has been involved in the application of microcomputers as electronic controllers.

(2) As part of his job duties at Rockwell he has been familiar with the Rockwell microcomputer known as the PPS-4 microcomputer. The PPS-4 microcomputer includes a microprocessor chip, one or more memory chips and one or more interface

chips. He has been involved in the design and development of PPS-4 microcomputer controllers for small machine systems since early 1978. Exhibit A hereto is a brochure dated 1973 describing the PPS-4 microcomputer and showing typical configurations of the PPS-4 microcomputer in Figures 1-5.

(3) He is familiar with a PPS-4 microcomputer controller module (hereafter the Gottlieb Controller) bearing assembly number PB00-D100. The controller module includes the PPS-4 microcomputer and, for use in pinball games, has control of operations of the lamps, solenoids and displays in response to action of the pinball playfield switches.

(4) He had responsibility for the development of a Rockwell microcomputer controller (designated the STC Controller Module) which was designed for, and suitable for, use as a general purpose industrial controller, the particular application of which was to be determined by the purchaser. It was intended to be used for such industrial control applications as laundry machine and dryer control, pharmaceutical pill making, controlling canning or bottling machines, garment folders, traffic controllers, printing equipment controllers, and for controlling banks of test equipment. Exhibit B hereto is a Preliminary Product Description for the STC Controller Module.

(5) The Rockwell STC Controller Module was formally announced by Rockwell at the Design Engineering Show in Chicago, Illinois in April, 1978, and other seminars on it were given.

Exhibit C hereto is a copy of a brochure entitled "STC Universal Controller Module" which was passed out to interested potential users at the Chicago show.

(6) A prototype STC Controller Module was built under his supervision at Rockwell. While the software was being developed and before any actual sale and delivery, the STC Controller program employing the PPS-4 microcomputer was discontinued in favor of an STC-type controller employing a faster operating Rockwell 6500 microcomputer. Nonetheless, the Rockwell STC Controller Module was suitable for use as a general purpose industrial controller.

(7) Exhibit D hereto is a copy of a photograph of a Gottlieb Controller. As is seen from the cover page of Exhibit C, the STC Controller Module appears identical to the Gottlieb Controller. In fact, except for the software programming stored in memory and for the need for larger memory, the STC Controller Module was intended to be identical to the Gottlieb Controller, even for industrial applications involving a greater degree of control than needed for pinball. Exhibit E is a copy of a letter which I wrote requesting that a slight, non-functional change be made to the then existing Gottlieb Controller to correspond to the STC Controller Module. Stated in other words, the STC Controller Module is a differently programmed Gottlieb Controller.

(8) Because of the similarity of the STC Controller Module and the Gottlieb Controller, I forwarded an actual

Gottlieb Controller to another Rockwell division for evaluation as to whether the STC Controller Module was suitable to be included in systems offered by that Rockwell division. Exhibit F is a copy of a letter identifying the Gottlieb Controller (No. P/N PB00-D100) as the STC Controller.

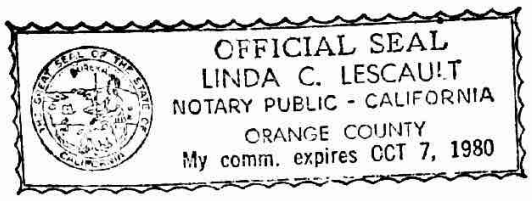
Further affiant sayeth not.

Zane A. Sandusky

SUBSCRIBED AND SWORN TO before me this 14th day of August, 1979.

Notary Public in and for
Orange County, California

[SEAL]



APPLICATION NOTES

PARALLEL PROCESSING SYSTEM (PPS)

Prepared by
W.E. Wickes

Microelectronics Division
Rockwell International

P.O. Box 3669
3430 Miraloma Ave.
Anaheim, California 92803
714/632-2859
TWX (910) 591-1179, Telex: Via TWX

12 bis, Rue Jean-Jaures
92-PUTEAUX-France
Telephone 7759000
Telex 61148

Akiyama Bldg.
25, Akefune-cho
Shiba-Nishikubo, Minato-ku
Tokyo 105 JAPAN
Phone: 501-9178
Telex J22198 Ans. Back: ROCKINT

518-D-17
April 1973
d in U.S.A.

INTRODUCTION

Rockwell Microelectronics has developed a compatible set of MOS/LSI circuits for use in building microprocessors. Processor is a general term used to describe the arithmetic and control section of a general purpose computer. Micro means small and that is just what has been developed: a small set of circuits which can be used to build equipment requiring digital data processing. This includes calculators, cash registers, credit terminals, electronic scales, billing machines, process controllers and general purpose data processors. We term this MOS/LSI circuit set, a "Parallel Processing System (PPS)."

PARALLEL PROCESSING SYSTEM

TABLE OF CONTENTS

ILLUSTRATIONS

Figure

TABLES

Tables

1. INTRODUCTION

Rockwell Microelectronics has developed a compatible set of MOS/LSI circuits known as the Parallel Processing System (PPS) for use in implementing a broad class of data processing products such as calculators, credit terminals, electronic cash registers, electronic weighing scales, billing machines, and process controllers.

The implementation of such a broad class of digital equipment is possible because of a one chip central processing unit (CPU) which receives and decodes an eight-bit instruction word to perform all required arithmetic and logic operations. The large number of instruction words to which the CPU responds gives it the capability of a 4-bit parallel minicomputer. These instruction words (referred to as microprograms) are stored in ROMs for rapid accessing while RAMs are used to provide register files and storage for working data. An input/output buffer and multiphase clock generator complete the basic circuits required to configure a total system. Figure 1 illustrates a typical organization while Table I lists the circuits currently available in the PPS family.

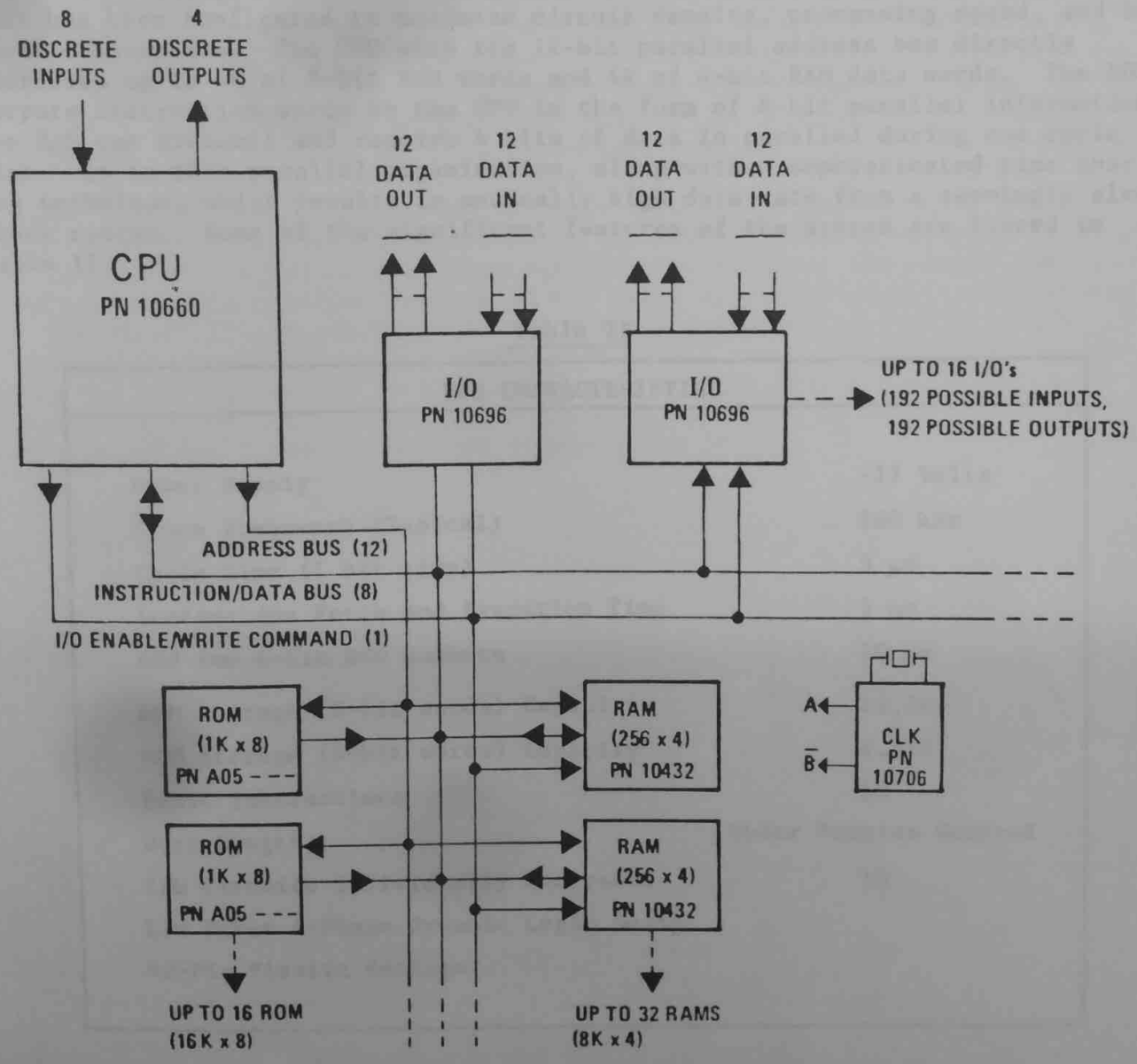


Figure 1. Basic System Configuration

Table I

PPS CIRCUITS

Part Number	Name	Description
10660	CPU	4-Bit Parallel Central Process Unit
10432	RAM	256 X 4 Random Access Memory
A05...	ROM	1024 X 8 Read Only Memory
A08...	ROM/RAM	Combination 704 X 8 ROM and 76 X 4 RAM
10696	I/O	Input/Output Buffer
10706	Clock	Multiphase Clock Generator

This set of circuits has been designed from scratch as a compatible group and each has been configured to maximize circuit density, processing speed, and low power consumption. The CPU with its 12-bit parallel address bus directly addresses up to 4K of 8-bit ROM words and 4K of 4-bit RAM data words. The ROM outputs instruction words to the CPU in the form of 8-bit parallel information. The RAM can transmit and receive 4 bits of data in parallel during one cycle time. It is this parallel organization, along with a sophisticated time sharing technique, which results in unusually high data rate from a seemingly slow clock system. Some of the significant features of the system are listed in Table II.

Table II

PPS CHARACTERISTICS

Power Supply	-17 Volts
Clock Frequency (Typical)	200 kHz
Cycle Time (1 bit time)	5 μs
Instruction Fetch and Execution Time	5 μs
Add two 4-bit BCD numbers	30 μs
ROM Storage (8-bit words) Capacity	16,384
RAM Storage (4-bit words) Capacity	8,192
Basic Instructions	50
Word Length	Under Program Control
I/O Circuits Individually Addressed	16
Low Power 4-Phase Dynamic Logic Design	
42-Pin Plastic Package	

The minimum PPS configuration possible is a 2-circuit implementation consisting of one CPU and one ROM/RAM as illustrated in Figure 2. (The clock chip is a low cost oscillator-driver circuit and is assumed with any given configuration.) This minimum organization is useful as a special purpose dedicated calculator or processor. It can receive 4-bit BCD data through four of the 8 discrete inputs and transmit 4-bit BCD data out through the four discrete outputs. The user can configure any combination of ROM, RAM and I/O circuits as may be required for a particular equipment implementation and can therefore trade off memory circuits to achieve the most cost-effective approach.

Several other examples are presented to convey various possible approaches. For instance, two of the A08 ROM/RAM circuits will provide 1408 words of ROM and 152 words of RAM while one A05 ROM and one 10432 RAM will provide 1024 words of ROM with 256 words of RAM. It can be seen that two A05 ROM with one 10432 RAM will provide more capability than three A08 ROM/RAM circuits. Any mix of memory circuits is allowed.

The I/O circuit is a general purpose time/impedance buffer between the MOS instruction/data bus and TTL logic signals. The CPU has an input/output instruction which will enable the transfer of data from one of three 4-bit input buffers in the I/O circuit to the accumulator in the CPU. Conversely, the I/O instruction can also accomplish transfer of data from the accumulator in the CPU to one of three 4-bit output registers in the I/O. Each I/O can be programmed with a unique 4-bit address allowing up to 16 separate I/O circuits to be attached to the instruction/data bus.

The clock circuit is designed to be operated with a 3.58 MHz low cost color TV crystal. Simply attaching this crystal to the proper pins and applying 17 volt power is all that is required to generate the proper output frequency, timing, and voltage signals required for any of the logic or memory circuits within the PPS. The clock circuit counts down this crystal frequency to provide a 200 kHz frequency output for clock output. This is the recommended mode of operation of the PPS and results in a 5 μs bit time or instruction execution time or cycle time. The addition of two 4-bit BCD numbers can be accomplished with 6 instructions and therefore will be completed in 30 μs.

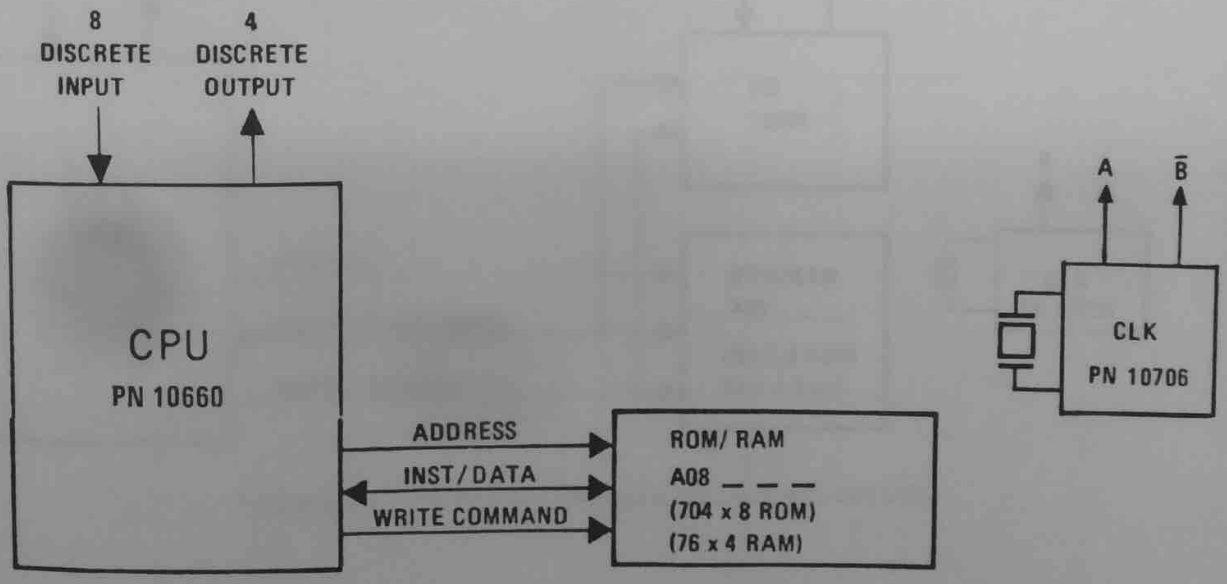


Figure 2. Minimum PPS Organization

2. EXAMPLES OF APPLICATIONS

Display Calculator Example -- A typical basic calculator organization is shown in Figure 3. This mechanization provides 704 words of microprogramming and 76 words of memory storage for working registers, display and memory registers. The I/O circuit provides 12 outputs and 12 inputs for strobing and encoding keyboards, driving segmented displays and sensing discrete switch settings. The CPU has 8 discrete inputs and 4 discrete outputs which interface directly with the accumulator, bringing the total number of interface pins to 20 inputs and 16 outputs. The unique characteristics of the PPS organization allow the user to emulate a ROM program before committing the microprogram to a MOS ROM, thereby guaranteeing product operation and performance prior to a volume procurement. In addition, the ability to readily add ROM, RAM and I/O circuits provides complete upward compatibility from the smallest product configuration shown in Figure 2 to one requiring a large amount of memory and interface. All internal signal levels between circuits in the PPS are MOS compatible impedance and voltages while all external data lines on the I/O are TTL compatible. A single crystal and + 5v, - 12v power supplies are all that is required to have a complete operating system.

The microprogram, for this type of product, is typically written such that the keyboard is periodically scanned in a predetermined sequence. The closure of a keyboard switch is a signal to the processor to carry out some operation or subroutine sequence as a function of the key which is depressed. This sequence of instructions is completely determined by the microprogram stored in the ROM.

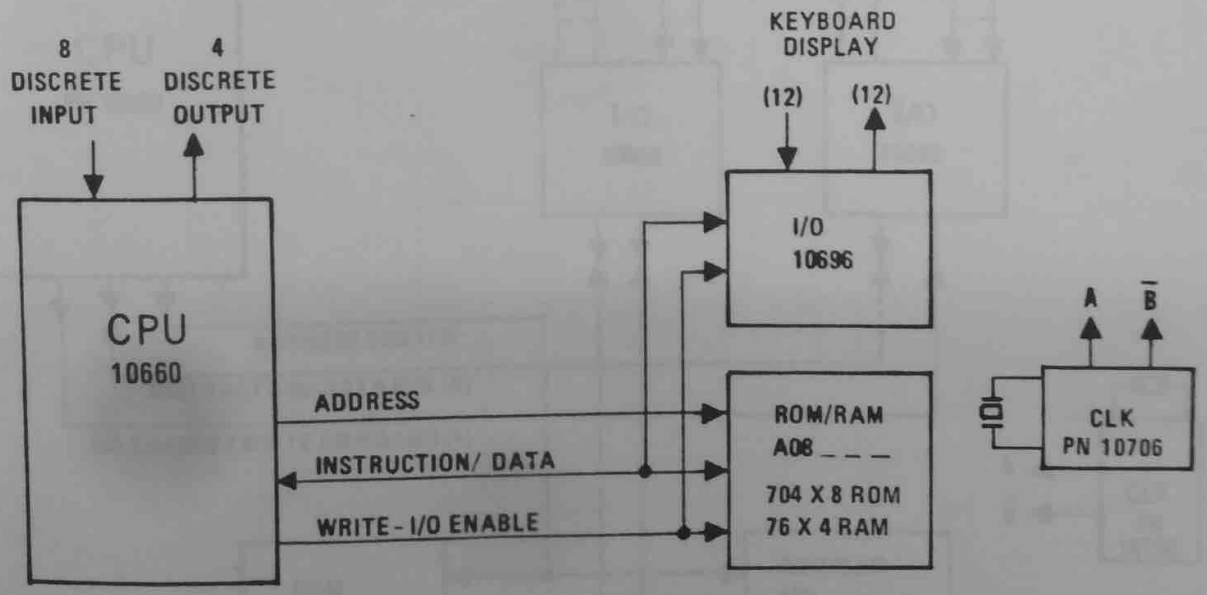


Figure 3. Display Calculator Organization

Printer-Display Calculator -- The ability of the PPS to handle a calculator with printer and display features is depicted by Figure 4. In this case, the control and formatting of data to be received by the printer has not only required an additional I/O circuit but also more ROM. Thus, the addition of the A05 ROM with the A08 ROM/RAM provides a total of 1728 words of ROM and 76 words of RAM. Typically, the second I/O is often required to provide sufficient interface for both a display and printer. These examples are primarily intended to show how a mix of the various available circuits can be used to achieve totally different products and not to imply that a printing calculator will require exactly those circuits illustrated. Obviously, the actual circuits required for a given application are determined only after a complete functional and operational specification has been established. The amount of microprogram required, for instance, is much less for a simple four function type of machine than would be required for a calculator designed to provide trigometric and logrithmetic features. The basic circuits would not change, however, just the amount of microprogram storage. This can be carried on into a fully programmable machine which would interface to a magnetic card input or cassette tape. It may be, where the volume of products justifies the cost, a custom designed I/O circuit can be designed to directly interface with the peripheral devices. Custom designed I/O circuits can usually provide all formatting, buffering, and timing required of peripheral devices such as printers, cassettes, and displays; except for current and voltage amplification; thus, greatly reducing the total component count required in a given design.

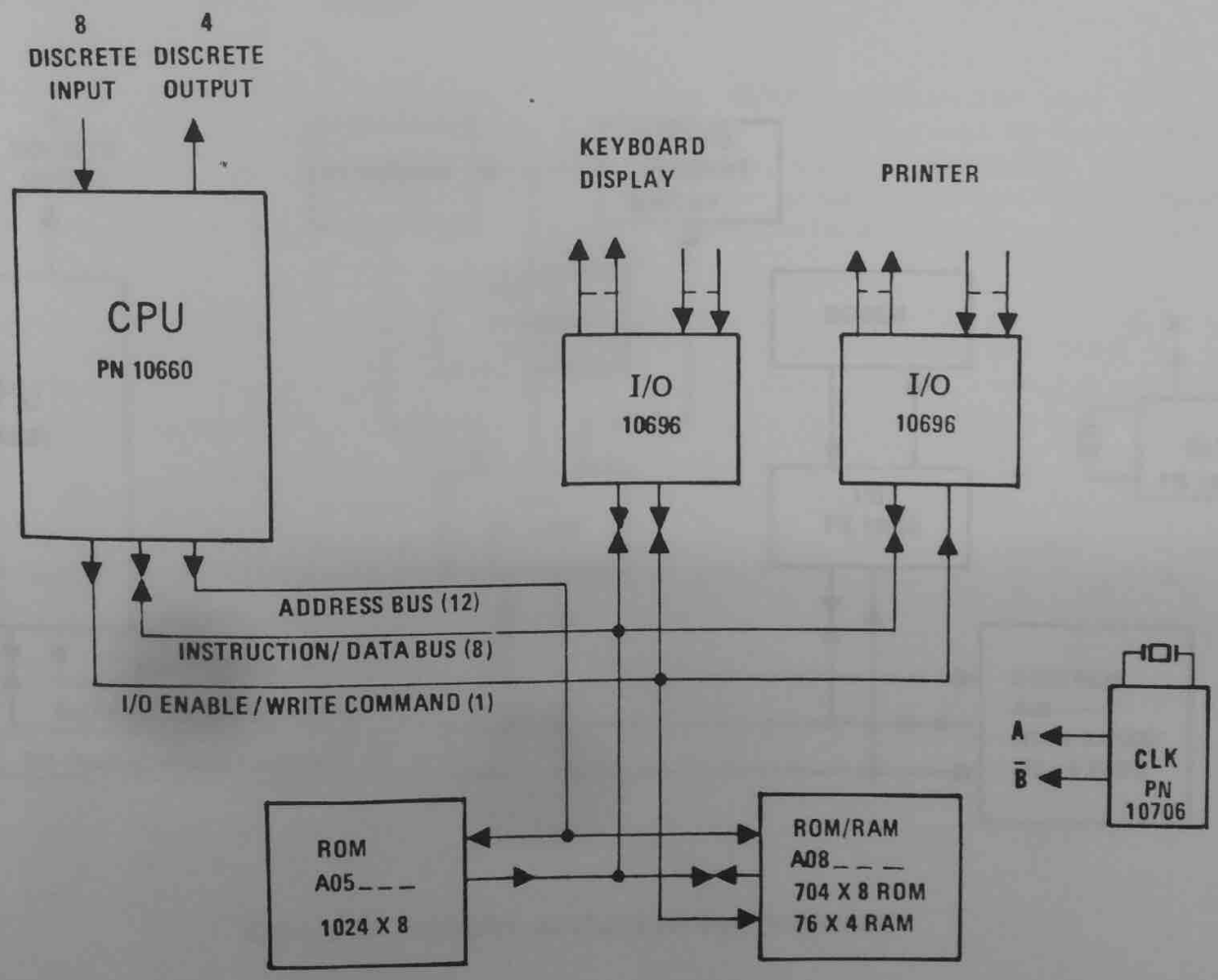


Figure 4. Printer - Display Calculator

Credit Authorization Terminal -- The implementation of a complete credit authorization terminal is illustrated in Figure 5. A credit authorization terminals' main function is to: a) collect data such as credit card, store number, terminal number, and amount of sale; b) transmit this data to a central processor either on-line or via a modem interface; and c) display resulting commands received from the central processor such as 'OK', 'HOLD', authorization number or such other instructions as may be programmed into the processor. Thus, it performs no arithmetic functions, just data transfer characteristics and therefore the amount of ROM and RAM is usually minimal. Figure 4 illustrates one way of expanding the actual number of keyboard and display strobe lines by using a TTL 1 of 16 decoder circuit. The 16 strobe lines can be used to form a matrix sampling feature with the four keyboard return lines such that up to 64 keys or switches can be sampled. These same 16 strobe lines can be used in conjunction with the 7 segment lines to multiplex 16 digits of display. The second I/O provides interface with a low speed modem for telephone line communication to a central computer. The modem circuit is not part of the PPS but can be designed by NRMEC in MOS to fit individual customer requirements. A simple custom MOS circuit would incorporate features suitable for directly interfacing between the PPS data lines and a telephone circuit; thereby reducing the total number of components required for the system.

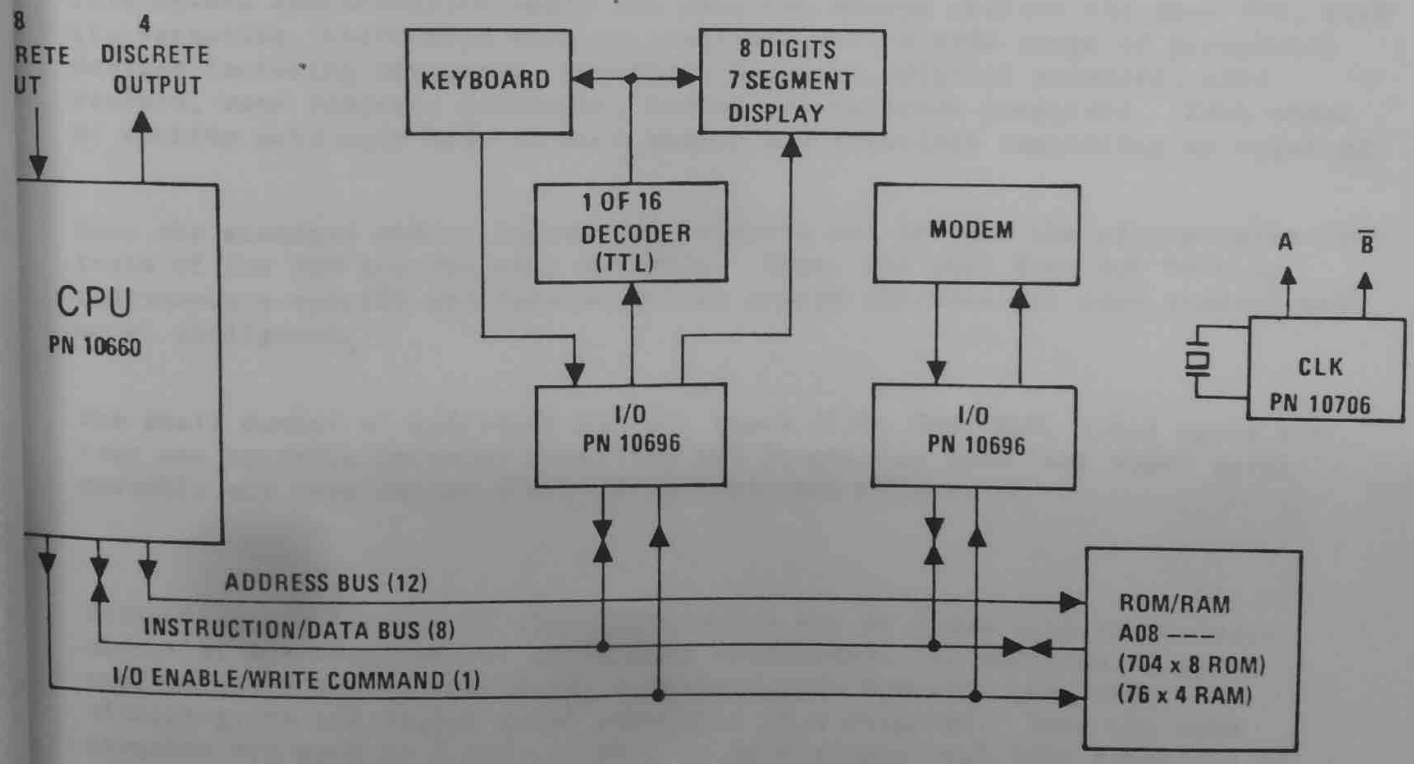


Figure 5. Credit Authorization Terminal

Summary -- The applications discussed are an indication of the flexibility and expandability of the PPS. The range of applications is limited primarily by the data handling rate.

The circuits (except for the clock) are packaged in 42-pin flat packs, providing maximum data interfacing capability.

The unique design of the CPU with its parallel data and address bus structures offers very high data handling and instruction execution rates. The 8-bit wide instruction format provides 256 unique instructions which can be created from the basic set of instructions. These instructions have all the logic, arithmetic and exchange routines normally found only in large general purpose computers.

The ability to address directly up to 4K of ROM and 4K of RAM provides full upward compatibility. That is, simple equipments may require only a few memory circuits while fully programmable machines may require the full complement of circuits. This range of machines can be designed by simply expanding on a fundamental set of routines and algorithms.

This upward compatibility means the user can always utilize the same CPU, with its versatile instruction set, to interface with a wide range of peripheral devices including keyboards, switches, displays, digital printers, card readers, wand readers, cassettes, modems and backroom computers. Each model of machine need only have as much memory and interface capability as required.

Once the standard and/or custom I/O circuits are defined the microprogram contents of the ROM are the only variable. Thus, the user does not have to continuously specify and receive unique custom circuits for each product and model configured.

The small number of different circuit types (CPU, ROM, RAM, I/Os) means the user can optimize incoming receiving and inspection such that board level assembly and test can be achieved with minimum effort.

Rockwell has a library of algorithms which can be drawn upon to minimize amount of ROM required for particular operations. In addition, a given configuration can be completely simulated with computer programs. The microprograms are always first simulated in a computer. Then the same programs are used in a ROM emulator to demonstrate real time operation of the designs.

3. BASIC SYSTEM OPERATION

The PPS has a repertoire of fifty basic instructions. These instructions control the operation of the PPS and are the key to the versatility of the systems.

During each bit time, the CPU will address the ROM, read and decode the instruction, execute the instruction, increment the ROM program counter, and load the RAM program counter in preparation for the next instruction. This bit time or single cycle instruction fetch and execution time is 5 μs. The proprietary architecture and multiphase clock timing techniques of the PPS result in this unusually high data handling rate from a relatively slow external clocking system.

System Timing -- The PPS circuits are controlled from a crystal controlled clock generator which provides two synchronized and phased clock signals. These signals, designated as A and $\bar{B}$, are received by the CPU and logically divided into four phases, such that the internal signals are being manipulated at four times the frequency of the A clock. For example, if the A clock is 200 kHz, logic signal flow within the CPU is occurring at 800 kHz.

The PPS parallel bus transfer lines are synchronized by the A and $\bar{B}$ clock signals such that data transfer occurs only during $\emptyset_2$ and $\emptyset_4$ time as depicted in Figure 6. During the alternate phase times, $\emptyset_1$ and $\emptyset_3$, the address and data bus lines are automatically cleared to zero. This unique interface timing enables the system to drive high capacitive loads normally associated only with larger and more complex systems. In fact, systems with up to 30 PPS devices can reliably share the PPS bus without the need for additional buffering or drive circuitry.

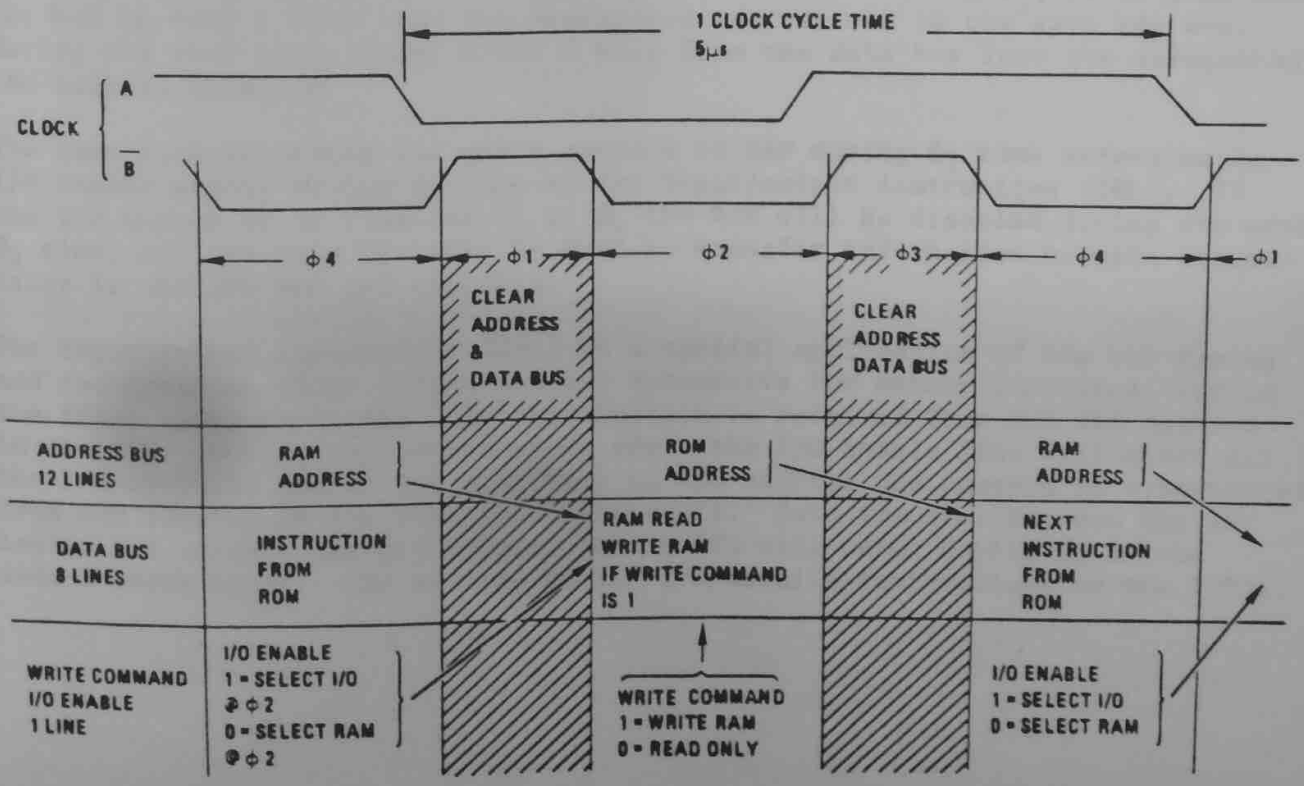


Figure 6. PPS Basic Bus Timing

Logic Levels -- A negative logic notation is used in the PPS system. That is, a logic one (1) is defined as the most negative voltage, while a logic zero (0) is defined as the most positive voltage. For example, assuming a - 17 volt power supply where V_{DD} = -17 volts and V_{SS} = 0 volts; a logic 0 is defined as more positive than - 1.9 volts while a logic 1 is defined as more negative than - 7.5 volts.

Multiplex System Data Transfer -- In addition to the power and clock signals there are 21 multiplexed lines interconnecting the CPU with ROM, RAM and I/O circuits. These lines, as shown in Fig 1, are functionally grouped as follows:

12 parallel address lines

8 parallel data lines

1 write command and I/O enable line.

The twelve address lines originate at the CPU and are time multiplexed within the CPU to provide direct addressing capability for up to 4096 locations on both the ROM and RAM. In addition to the twelve direct address lines, the A05 ROM circuit has two chip select inputs and the 10432 RAM circuit has one chip select input. These chip select lines may be directly controlled by discrete outputs from CPU or I/O circuits for memory expansion without the need for auxiliary circuitry. Memory expansion is more fully explained in the description of the ROM and RAM.

Like the address lines the eight data bus lines are time-shared lines from the CPU. During $\emptyset_2$ a logical "one" on the write/I-O line interpreted by the RAMs as a write enable command and data on the bus will be written in to RAM. The RAM is a non-destructive read-out device and, therefore, is always programmed to read; however, it must be instructed to "write". Because the 8 data lines are functioning as a dual 4-bit bidirection data bus during $\emptyset_2$, it is possible for RAM to read 4 bits from the designated address out to the data bus and, during the same cycle time, write 4 bits from the data bus into the designated RAM address location.

The same line providing the write command to RAM during $\emptyset_2$ time serves as an I/O enable signal during $\emptyset_4$ time of the input/output instruction (IØL). If the I/O enable is on (logical 1) at $\emptyset_4$ the RAM will be disabled during the next $\emptyset_2$ time, and the data bus will be used to transfer information between accumulator in the CPU and I/O circuits.

The input/output instruction (IØL) is a special application of the bus timing and requires two clock cycles and two successive ROM memory locations. During the first clock cycle the "IØL" instruction is received from ROM and decoded in the CPU. During the second clock cycle the I/O enable line will alert all the I/O circuits and at the same time an I/O address and command is transmitted from ROM turning on the selected I/O circuit. Data transfer between the selected I/O circuit and accumulator in the CPU will occur during $\emptyset_2$ of the second clock cycle. The section on the I/O details instructions to the I/O's.

4. CIRCUIT DESCRIPTIONS

A. 10660 CPU -- The 10660 Central Process Unit (CPU) is a 4-bit parallel processor implemented with four phase dynamic logic for operation from a single 17 volt power supply. The CPU, through a unique time sharing system, utilizes an 8-bit parallel data bus to transfer 8-bit word instructions from ROM to CPU and also as a dual bidirectional 4-bit parallel data bus to transfer data between the CPU and RAMs and I/Os.

The CPU contains: (a) logic necessary to receive and decode 50 basic instructions; (b) 4-bit parallel adder-accumulator for arithmetic and logic operations; (c) 12-bit counter for creating and storing addresses for RAM; (d) 12-bit counter and two 12-bit save registers for creating and storing addresses for ROM; (e) 8 discrete input synchronizers; (f) 4 discrete output static drivers; (g) two individual control flip-flops and (h) multiplexed receivers and drivers for interfacing with the 12-bit parallel address bus and the 8-bit parallel data/instruction bus. A block diagram of the CPU is shown in Figure 7.

(a) Micro Instruction Decode -- The decode portion of the chip contains logic to decode the instructions and provide signals to control data transfers, arithmetic operations and logical sequences. Instructions are coded in one byte (8 bits) and grouped as arithmetic, logical, data transfer, control transfer, input/output and special instructions. Transfer and Mark, Transfer Long, Load Base Long, Load Base and Input/Output are considered long instructions since they require two bytes. All instructions are executed in one cycle time of the "A" clock except long and indirect instructions and the I/O instruction which are executed in two cycle times. A detail listing of instructions to which the CPU will respond is contained in Table III.

(b) Adder-Accumulator -- The adder is a 4-bit parallel binary adder and carry flip-flop interconnected with the accumulator A. One instruction (AD) will accomplish the addition of the accumulator with the contents of a designated memory location. The EX instruction performs the exchange of 4-bits of data between the accumulator and a designated location in RAM during one cycle time. The control logic within the CPU is such that arithmetic or logical instructions are carried out in one cycle time. The addition of two decimal digits can be carried out with six instructions or 6 cycle times; thus, assuming a 5 μs cycle time (200 kHz clock), the time to add two decimal digits is 30 μs. The subtraction of two decimal digits can also be accomplished within six instructions and, thus, within 6 cycle times or 30 μs.

(c) RAM address Register -- The 12-bit B register, consisting of three 4-bit registers (BU, BM and BL), contains the next address location to be accessed in RAM. Register BL can be incremented or incremented by 1 through program control providing the ability to sequence through RAM registers serially by 4-bit character. The upper 8 bits (BU and BM) of RAM address can be set to special values by program control during a transfer operation of data to or from a distant RAM location into the working storage area indicated by the 4-bits of BL. During RAM address modification operations it is also possible to use accumulator A and register X for temporary storage of the upper 8 bits of the new address retrieval from RAM without destroying the current RAM address pointer.

(d) Program Counter and Save Register -- The P counter contains the address of the next microinstruction and is automatically decremented each cycle time during normal operation. The SA and SB registers are two "Save" registers which provide a two level stack for holding instruction addresses during branching. This gives a direct two level nesting capability for storing pro-

gram subroutine addresses in hardware; however, unlimited nesting is available via the CYS instruction. This instruction recirculates register SA and accumulator A and, therefore, provides for saving ROM microprogram addresses in RAM. Since RAM can be used for storing microprogram addresses, subroutine nesting becomes limited only by the amount of dedicated storage allocated for this purpose.

(e) Control FF and Discrete I/O -- There are two control flip-flops, FF1 and FF2, which can be set, reset and tested by program control. There are, also, 8 discrete inputs which can be copied directly into the accumulator as two separate 4-bit groups. These groups are identified as discrete inputs group A (DIA) and discrete inputs group B (DIB). These inputs are intended to be used for inputing special signals, switch positions or flags from circuitry external to the PPS. The contents of the accumulator can be outputted directly to circuits external to the PPS via a special 4-bit parallel output register (DOA) with individual output drivers.

(f) Power on Reset -- During power turn-on it is necessary to initialize the CPU to a known state such that the proper sequence of events can occur. This initialization is accomplished by input of a Power-on signal which is generated external to the PPS. The CPU receives this signal, initializes the internal logic states and, at the same time, generates a synchronized Power-on output signal which can be used to initialize other circuits within the PPS.

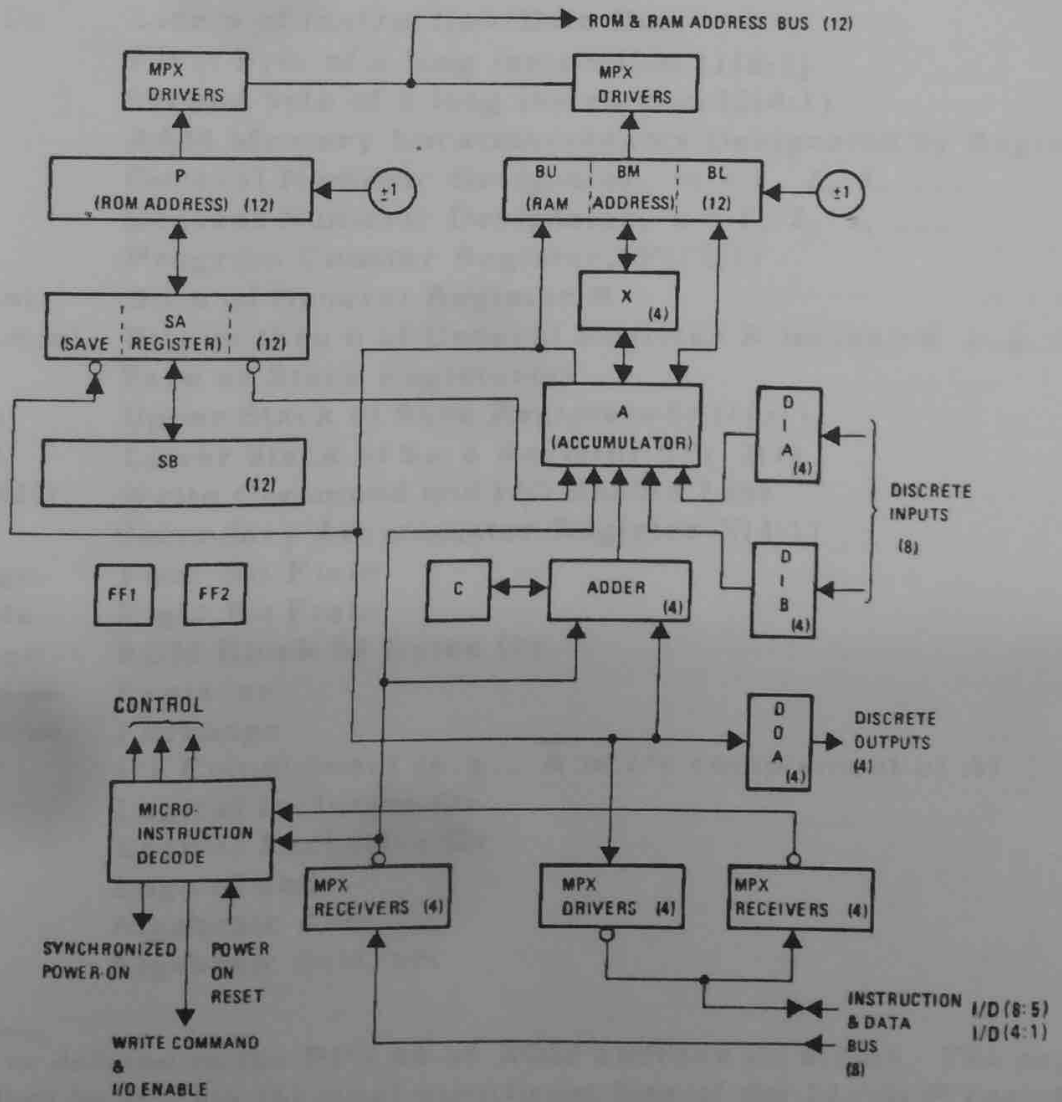


Figure 7. 10660 CPU Block Diagram

Instruction List --Table IV provides a listing of the 50 instructions which can be used to control generation of ROM and RAM address as well as manipulation and transfer of data between the CPU and RAM and I/O. Definitions of symbology used in the instruction list are also provided such that a programmer can easily understand the instruction list and begin to visualize how the PPS may be used for a given application.

Table III
INSTRUCTION NOMENCLATURE

Symbol	
A	Accumulator Register, A(4:1)
A/Bn	Line n of Address Bus
B	RAM Address Register, B(12:1)
C	Carry Link Flip-Flop
FF1	General Flip-Flop Number 1
FF2	General Flip-Flop Number 2
I	Instruction I(8:1)
I/Dn	Line n of Instruction/Data Bus
I1	First byte of a long instruction I1(8:1)
I2	Second byte of a long instruction I2(8:1)
M	RAM Memory Location contents Designated by Register B
m	General Numeric Designator, m = 1, 2, 3, ...
n	General Numeric Designator, n = 1, 2, 3, ...
P	Program Counter Register, P(12:1)
R(n)	Bit n of General Register R
R(m:n)	Bits m thru n of General Register R inclusive e.g. R(12:7)
S	Save or Stack Register(s)
SA	Upper Stack of Save Registers SA(12:1)
SB	Lower Stack of Save Register SB(12:1)
W/IO	Write Command and I/O Enable Line
X	Secondary Accumulator Register X(4:1)
Digit	Four Bit Field
Byte	Eight Bit Field
Page	ROM Block 64 Bytes (*)
←	Replaces
↔	Exchange
$\overline{\quad}$	1's Complement (e.g., $\overline{A}$ is 1's complement of A)
V	Logical Inclusive Or
⊻	Logical Exclusive Or
∧	Logical and
+	Algebraic Add
−	Algebraic Subtract

*A page is defined in the PPS as 64 ROM address locations, The page number is specified by the six (6) most significant bits of the 12-bit P register. The locations within a page are defined by the six (6) least significant bits.

ARITHMETIC INSTRUCTIONS

Mnemonics	I/D Bus OP Code Hex & Binary	Name	Description	Symbolic Equation
AD	OB 0000 1011	Add (1 cycle)	The result of binary addition of contents of accumulator and 4-bit contents of the RAM currently addressed by B register, replaces the contents of accumulator. The resulting carry-out is loaded into C flip-flop.	C, A ← A+M
ADC	OA 0000 1010	Add with carry-in (1 cycle)	Same as AD except the C flip-flop serves as a carry-in to the adder.	C,A ← A+M+C
ADSK	O9 0000 1001	Add and skip on carry-out (1 cycle)	Same as AD except the next ROM word will be skipped (ignored) if a carry-out is generated.	C, A ← A+M Skip if C = 1
ADCSK	O8 0000 1000	Add with carry-in and skip on carry-out (1 cycle)	Same as ADSK except the C flip-flop serves as a carry-in to the adder.	C, A ← A+M+C Skip if C = 1
ADI	60-6E *0110 xxxx Except 65	Add immediate and skip on carry-out (1 cycle)	The result of binary addition of contents of accumulator and 4-bit immediate field of instruction word replaces the contents of accumulator. The next ROM word will be skipped (ignored) if a carry-out is generated. This instruction does not use or change the C flip-flop. The immediate field I(4:1) of this instruction may not be equal to binary 0000 or 1010. (See CYS and DC)	A ← A+|I(4:1)| Skip if carry-out = one I(4:1) ≠ 0000 I(4:1) ≠ 1010 See Note 3
DC	65 0110 0101	Decimal Correction (1 cycle)	Binary 1010 is added to contents of accumulator. Result is stored in accumulator. Instruction does not use or change carry flip-flop or skip.	A ← A+1010

LOGICAL INSTRUCTIONS

AND	OD 0000 1101	Logical AND (1 cycle)	The result of logical AND of accumulator and 4-bit contents of RAM currently addressed by B register replaces contents of accumulator.	A ← A∧M
OR	OF 0000 1111	Logical OR (1 cycle)	The result of logic OR of accumulator and 4-bit contents of RAM currently addressed by B register replaces contents of accumulator.	A ← A∨M
EOR	OC 0000 1100	Logical Exclusive-OR (1 cycle)	The result of logic exclusive-OR of accumulator and 4-bit contents of RAM currently addressed by B register replaces contents of accumulator.	A ← A∀M
COMP	OE 0000 1110	Complement (1 cycle)	Each bit of the accumulator is logically complemented and placed in accumulator.	A ← $\bar{A}$

*xxxx Indicates restrictions on bit patterns allowable in immediate field as specified in the symbolic equation description.

DATA TRANSFER INSTRUCTIONS

Mnemonics	I/D Bus OP Code Hex & Binary	Name	Description	Symbolic Equation
SC	20 0010 0000	Set Carry flip-flop (1 cycle)	The C flip-flop is set to 1.	C ← 1
RC	24 0010 0100	Reset Carry flip-flop (1 cycle)	The C flip-flop is set to 0.	C ← 0
SF1	22 0010 0010	Set FF1 (1 cycle)	Flip-flop 1 is set to 1.	FF1 ← 1
RF1	26 0010 0110	Reset FF1 (1 cycle)	Flip-flop 1 is set to 0.	FF1 ← 0
SF2	21 0010 0001	Set FF2 (1 cycle)	Flip-flop 2 is set to 1.	FF2 ← 1
RF2	25 0010 0101	Reset FF2 (1 cycle)	Flip-flop 2 is set to 0.	FF2 ← 0
LD	30-37 0011 0 . . .	Load Accumulator from Memory (1 cycle)	The 4-bit contents of RAM currently addressed by B register are placed in the accumulator. The RAM address in the B register is then modified by the result of an exclusive-OR of the 3-bit immediate field I(3:1) and B(7:5).	A ← M; B(7:5) ← B(7:5) ∀ |I(3:1)| See Note 3
EX	38-3F 0011 1 . . .	Exchange Accumulator and Memory (1 cycle)	Same as LD except the contents of accumulator are also placed in currently addressed RAM location.	A ↔ M B(7:5) ← B(7:5) ∀ |I(3:1)| See Note 3
EXD	28-2I 0010 1 . . .	Exchange Accumulator and Memory and decrement BL (1 cycle) See Note 3	Same as EX except RAM address in B register is further modified by decrementing BL by 1. If the new contents of BL is 1111, the next ROM word will be ignored.	A ↔ M B(7:5) ← B(7:5) ∀ |I(3:1)|; BL ← BL-1 Skip on BL=1111
LDI	70-7I 0111	Load Accumulator Immediate (1 cycle)	The 4-bit contents, immediate field I(4:1), of the instruction are placed in accumulator. (See Note below)	A ← |I(4:1)| See Note 3
LAX	12 0001 0010	Load Accumulator from X register (1 cycle)	The 4-bit contents of the X register are placed in the accumulator.	A ← X
LXA	1B 0001 1011	Load X Register from Accumulator (1 cycle)	The contents of the accumulator are transferred to the X register.	X ← A
LABL	11 0001 0001	Load Accumulator with BL (1 cycle)	The contents of BL register are transferred to the accumulator.	A ← BL
LBMX	10 0001 0000	Load BM with X (1 cycle)	The contents of X register are transferred to BM register.	BM ← X
LBUA	04 0000 0100	Load BU with A (1 cycle)	The contents of accumulator are transferred to BU register. Also, the contents of currently addressed RAM are transferred to accumulator.	BU ← A ← M

NOTE
Only the first occurrence of an LDI in a consecutive string of LDI's will be executed. The program will ignore the remaining LDI's and execute next valid instruction.

DATA TRANSFER INSTRUCTIONS (CONT)

Mnemonics	I/D Bus OP Code Hex & Binary	Name	Description	Symbolic Equation
XABL	19 0001 1001	Exchange Accumulator and BL (1 cycle)	The contents of accumulator and BL register are exchanged.	A ↔ BL
XBMX	18 0001 1000	Exchange BM and X (1 cycle)	The contents of BM register and X register are exchanged.	X ↔ BM
XAX	1A 0001 1010	Exchange Accumulator and X (1 cycle)	The contents of accumulator and X register are exchanged.	A ↔ X
XS	06 0000 0110	Exchange SA and SB (1 cycle)	The 12-bit contents of SA register and SB register are exchanged.	SA ↔ SB
CYS	6F 0110 1111	Cycle SA register and accumulator. (1 cycle)	A 4-bit right shift of the SA register takes place with the four bits which are shifted off the end of SA being transferred into the accumulator. The contents of the accumulator are placed in the left end of SA register.	A ← SA(1:1) SA(4:1) ← SA(8:5) SA(8:5) ← SA(12:9) SA(12:9) ← A
LB	CO-CF 1st word 1100 .... 2nd word from page 3 ---- ----	Load B Indirect (2 cycles)	Sixteen consecutive locations on ROM page 3 (I_2) contain data which can be loaded into the eight least significant bits of the B register by use of any LB instruction. The four most significant bits of B register will be loaded with zeros. The contents of the SB register will be destroyed. This instruction takes two cycles to execute but occupies only one ROM word. (Automatic return) (See Note below.)	SB ← SA ← P P(12:5) ← 0000 0011 P(4:1) ← I1(4:1) BU ← 0000 B(8:1) ← |I2(8:1)| P ← SA ↔ SB See Notes 3 and 4
LBL	00 1st word 0000 0000 2nd word ---- ----	Load B Long (2 cycles)	This instruction occupies two ROM words, the second of which will be loaded into the eight least significant bits of the B register. The four most significant bits of B (BU) will be loaded with zeros. (See Note below)	BU ← 0000 B(8:1) ← |I2(8:1)| See Note 3
INCB	17 0001 0111	Increment BL (1 cycle)	BL register (least significant four bits of B register) is incremented by 1. If the new contents of BL is 0000, then the next ROM word will be ignored.	BL ← BL+1 Skip on BL=0000
DECB	1F 0001 1111	Decrement BL (1 cycle)	BL register is decremented by 1. If the new contents of BL is 1111, then the next ROM word will be ignored.	BL ← BL-1 Skip on BL = 1111

NOTE

Only the first occurrence of an LB or LBL instruction in a consecutive string of LB or LBL will be executed. The program will ignore the remaining LB or LBL and execute the next valid instruction. Within subroutines the LB instruction must be used with caution because the contents of SB have been modified.

CONTROL TRANSFER INSTRUCTIONS

Mnemonics	I/D Bus OP Code Hex & Binary	Name	Description	Symbolic Equation
T	80-BF 10-- ----	Transfer (1 cycle)	An unconditional transfer to a ROM word on the current page takes place. The least significant 6-bits of P register P(6:1) are replaced by six bit immediate field I(6:1).	P(6:1) ← I(6:1)
TM	DO-FF * 1st word 11xx ---- 2nd word from page 3 ---- ----	Transfer and Mark Indirect (2 cycles)	48 Consecutive locations on ROM page 3 contains pointer data which identify subroutine entry addresses. These subroutine entry addresses are limited to pages 4 through 7. This TM instruction will save the address of the next ROM word in the SA register after loading the original contents of SA into SB. A transfer then occurs to one of the subroutine entry addresses. This instruction occupies one ROM word but takes two cycles for execution.	SB ← SA ← P P(12:7) ← 000011 P(6:1) ← I1(6:1) P(12:9) ← 0001 P(8:1) ← I2(8:1) See Note 4 Note: I1(6:5) ≠ 00
TL	50-5F 1st word 0101 ---- 2nd word ---- ----	Transfer Long (2 cycles)	This instruction executes a transfer to any ROM word on any page. It occupies two ROM words and requires two cycles for execution. The first byte loads P(12:9) with field I1(4:1) and then the second byte I2(8:1) is placed in P(8:1).	P(12:9) ← I1(4:1); P(8:1) ← I2(8:1)
TML	01-03 * 1st word 0000 00xx 2nd word ---- ----	Transfer and Mark Long (2 cycles)	This instruction executes a transfer and mark to any location on ROM pages 4 through 15. It occupies two ROM words and requires two cycle times for execution.	See Note 4 SB ← SA ← P P(12:9) ← I1(4:1) P(8:1) ← I2(8:1) Note I1(2:1) ≠ 00
SKC	15 0001 0101	Skip on Carry flip-flop (1 cycle)	The next ROM word will be ignored if C flip-flop is 1.	Skip if C = 1
SKZ	1E 0001 1110	Skip on Accumulator Zero (1 cycle)	The next ROM word will be ignored if accumulator is zero.	Skip if A = 0
SKBI	40-4F 0100 ----	Skip if BL Equal to Immediate (cycle)	The next ROM word will be ignored if the least significant four bits of B register (BL) is equal to the 4-bit immediate field I(4:1) of instruction.	Skip if BL = I(4:1)
SKF1	16 0001 0110	Skip if FF1 Equals 1 (1 cycle)	The next ROM word will be ignored if FF2 is 1.	Skip if FF1 = 1
SKF2	14 0001 0100	Skip if FF2 Equals 1 (1 cycle)	The next ROM word will be ignored if FF1 is 1.	Skip if FF2 = 1
RTN	05 0000 0101	Return (1 cycle)	This instruction executes a return from subroutine by loading contents of SA register into P register and interchanges the SB and SA registers.	P ← SA ↔ SB

*xx Indicates restrictions on bit patterns allowable in the designated bit positions in the instruction field as specified in the symbolic equation description.

CONTROL TRANSFER INSTRUCTIONS (CONT)

Mnemonics	I/D Bus OP Code Hex & Binary	Name	Description	Symbolic Equation
RTNSK	07 0000 0111	Return and Skip (1 cycle)	Same as RTN except the first ROM word encountered after the return from subroutine is skipped.	P ← SA ↔ SB P ← P+1

INPUT/OUTPUT INSTRUCTIONS

Mnemonics	I/D Bus OP Code Hex & Binary	Name	Description	Symbolic Equation
IOL	1C 1st word 0001 1100 2nd word ---- ----	Input/Output Long (2 cycles)	This instruction occupies two ROM words and requires two cycles for execution. The first ROM word is received by the CPU and sets up the I/O Enable signal. The second ROM word is then received by the I/O devices and decoded for address and command. The contents of the accumulator inverted are placed on the data lines for acceptance by the I/O. At the same time, input data received by the I/O device is transferred to the accumulator inverted.	A → Data Bus A ← $\overline{\text{Data Bus}}$ I2 → I O Device
DIA	27 0010 0111	Discrete Input Group A (1 cycle)	Data at the inputs to discrete. Group A is transferred to the accumulator.	A ← DIA
DIB	23 0010 0011	Discrete Input Group B (1 cycle)	Data at the inputs to discrete. Group B is transferred to the accumulator.	A ← DIB
DOA	1D 0001 1101	Discrete Output (1 cycle)	The contents of the accumulator are transferred to the discrete output register.	D O A ← A

SPECIAL INSTRUCTION

SAG	13 0001 0011	Special Address Generation (1 cycle)	This instruction causes the eight most significant bits of the RAM address output to be zeroed during the next cycle only. Note that this instruction does not alter the contents of the B register.	A/B Bus (12:5) ← 0000 0000 A/B Bus (4:1) ← BL(4:1) Contents of "B" remain unchanged

GENERAL NOTES

(1) The word "skip" or "ignore" as used in this instruction set means the instruction will be read from memory but not executed. Each skipped or ignored word will require one clock cycle time.

(2) The reference to ROM pages and locations are defined as the ROM address appearing on the A/B bus. During initial Power On the starting address is Page 0 Location 0 and is automatically incremented each clock cycle.

(3) Instruction ADI, LD, EX, EXD, LDI, LB and LBL have a numeric value coded as part of the instruction in the immediate field. This numeric value must be in complementary form on the bus. All of these immediate fields which are inverted are shown in brackets.

For example: ADI 1, as written by the programmer who wishes to add one to the value in the accumulator, is converted to $6E_{(16)}$=0110 [1110]; the bracketed binary value is the value as seen on the data bus. If the programmer is using the Rockwell Assembler he does not have to manually determine the proper inverted value as the assembler does this for him.

(4) On all instructions which transfer the contents of P into SA, the P register has already been advanced to the next instruction location.

A05-ROM -- The A05--ROM shown in Figure 8 is a 8192-bit Read Only Memory anized in a 1024x8 bit configuration. It has been designed with dynamic ress decode logic for operation from a single 17 volt power supply, "A" and clock inputs, a maximum access time of 1.9 μs, and for interface compati- ity with the PPS 10660 CPU. It is intended to be used to store 8-bit roinstruction routines for control and sequencing of data within a PPS set.

10660 CPU, with its 12-bit parallel address outputs, can directly address to 4 ROM chips or 4096 words of microinstruction. The ROM, with its 10 ress and 4 chip select inputs, provides for direct selection of up to 16 chips or 16,384 words of microinstruction. The two additional chip select uts on the ROM chip (which cannot be addressed by the 12-bit address bus) be addressed by outputs from an I/O circuit or the discrete outputs avail- e on the CPU.

unique time-sharing design of the PPS address and data bus is such that ormation on the data bus is treated as 8 bits of microinstruction during one k phase time and as dual 4-bit data buses during the opposite phase time. s precludes the ROM from being used for storage of permanent data such as t-up tables during the normal ROM Clock phase time; however, the A05 ROM is gned such that it can be encoded to respond to the clock phase time allo- d to the RAM. During the RAM phase time the RAM can receive 4 bits of from one of the 4-bit parallel data buses and transmit 4 bits of data to other 4-bit parallel data bus. Thus, by encoding the ROM to respond to the ess and data buses during the RAM phase time, it can be used to transmit

ts of data onto the data bus. In this configuration, the 8 parallel out- from the ROM can be externally connected to form a 4-bit parallel output which time control input A11X is used to select the upper or lower 4 bits he 8-bit output. This has the effect of creating a 2048x4 bit organiza- for storage of permanent data which is addressed and accessed during the phase time.

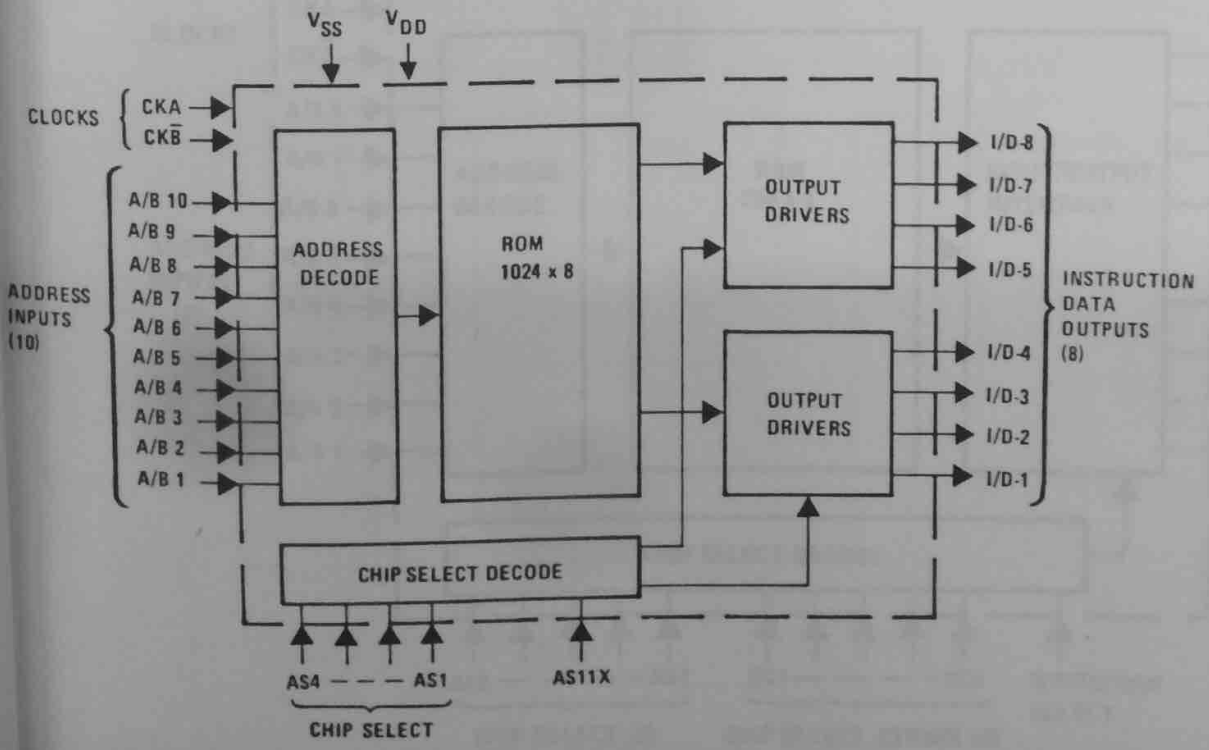


Figure 8. A05 -- ROM (1024x8) Block Diagram

C. 10432 RAM -- The 10432 RAM shown in Figure 9 is a 1024-bit Random Access Memory organized in a 256x4 bit configuration. It has been designed as a dynamic memory with automatic refresh logic for operation from a single 17 volt power supply, "A" and "$\bar{B}$" clock inputs, a maximum access time of 1.9 μs and for interface compatibility with the PPS 10660 CPU. It is intended to be used as a general working and memory register storage of data within a PPS set.

The 10660 CPU, with its 12-bit parallel address outputs can directly address up to 16 RAM chips or 4096 words of 4-bit data. The RAM, with its 8 parallel address and 5 chip select inputs, provides for direct selection of up to 32 RAM chips or 8192 words of data. The one additional chip select input can be addressed by an output from an I/O circuit or a discrete output from the CPU.

The unique time sharing design of the PPS address and data buses is such that the contents of a given location in RAM can be exchanged with the contents of the accumulator in the CPU during one clock phase time. This is possible because when the RAM is addressed it shifts the 4-bit data contents of the addressed location to the RAM output registers which place the data on one of the 4-bit parallel data buses. At the same time, if the write/RAM select input to the RAM is enabled, the RAM captures the data on the other 4-bit parallel data bus and copies it into the memory location being addressed. In order for this to work, the CPU has placed the contents of the accumulator into its output register (which is the data being written into the RAM) and copies the data output from the RAM into the accumulator. This timing and data exchange capability is a major key to the high data rate and predictable performance of the PPS.

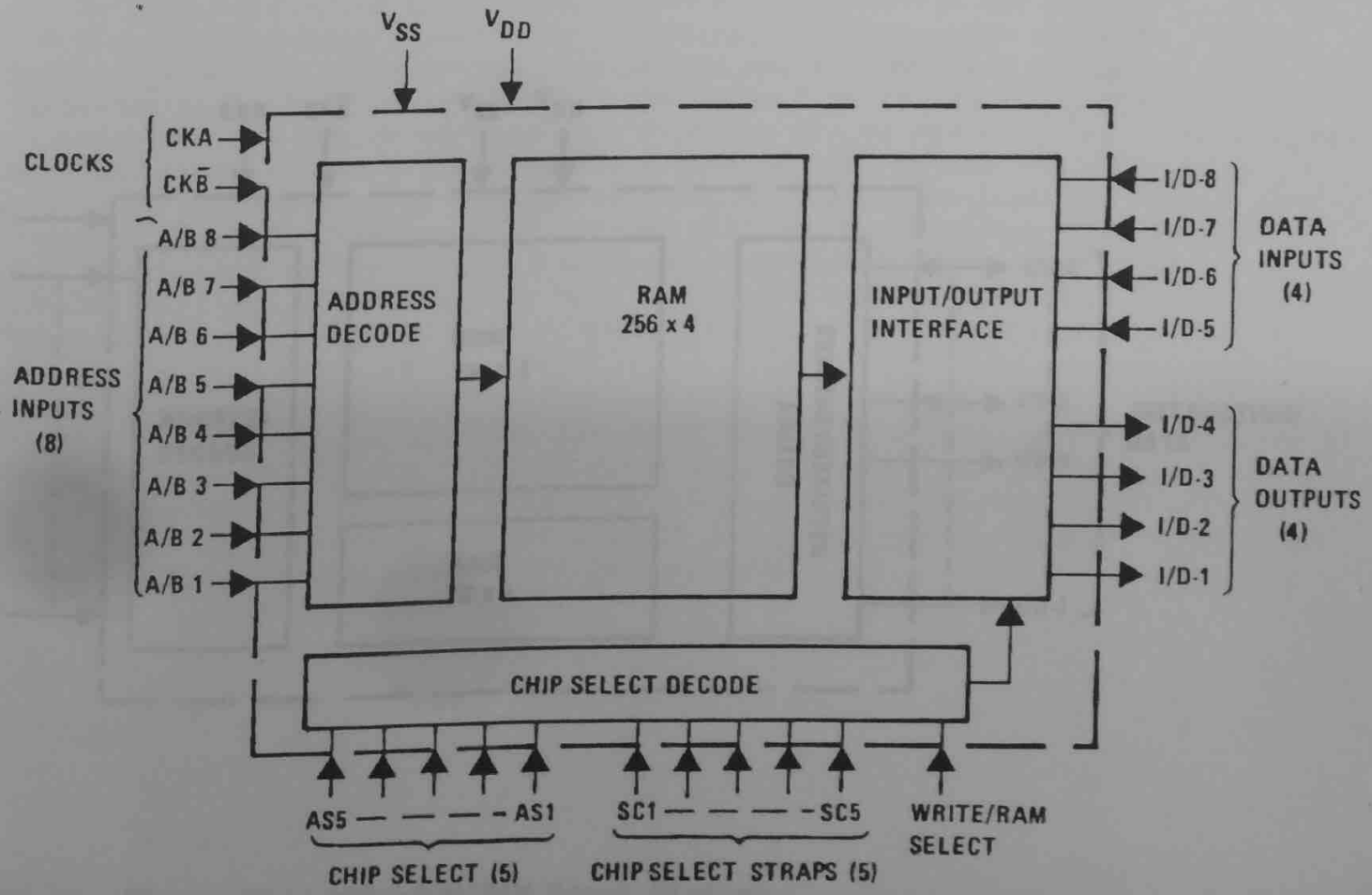


Figure 9. 10432 - RAM (256x4) Block Diagram

A08 - ROM/RAM -- The A08 ROM/RAM, shown in Figure 10, is a circuit ntaining a mask programmable MOS ROM as well as a RAM. It is uniquely signed to interface with the PPS CRU address and data bus and, as such, is eful in equipments requiring only a relatively small amount of ROM and M. The ROM section is organized as 704 x 8 words (5,632 bits) while the M section is organized as 76 x 4 words (304 bits). The chip interfaces th the common address bus and contains a single address decoder section. ternal clock timing is generated such that the decoded address is routed the RAM section or the ROM section as a function of "A" and "$\overline{B}$" clock puts. In a like manner, the outputs drive or receive data from the 8-bit struction/data bus in the PPS as a function of clock timing and control gnals. From the users' point of view, this single chip provides the same nctions as the A08 ROM chip and the 10432 RAM chip except for the smaller ount of ROM and RAM.

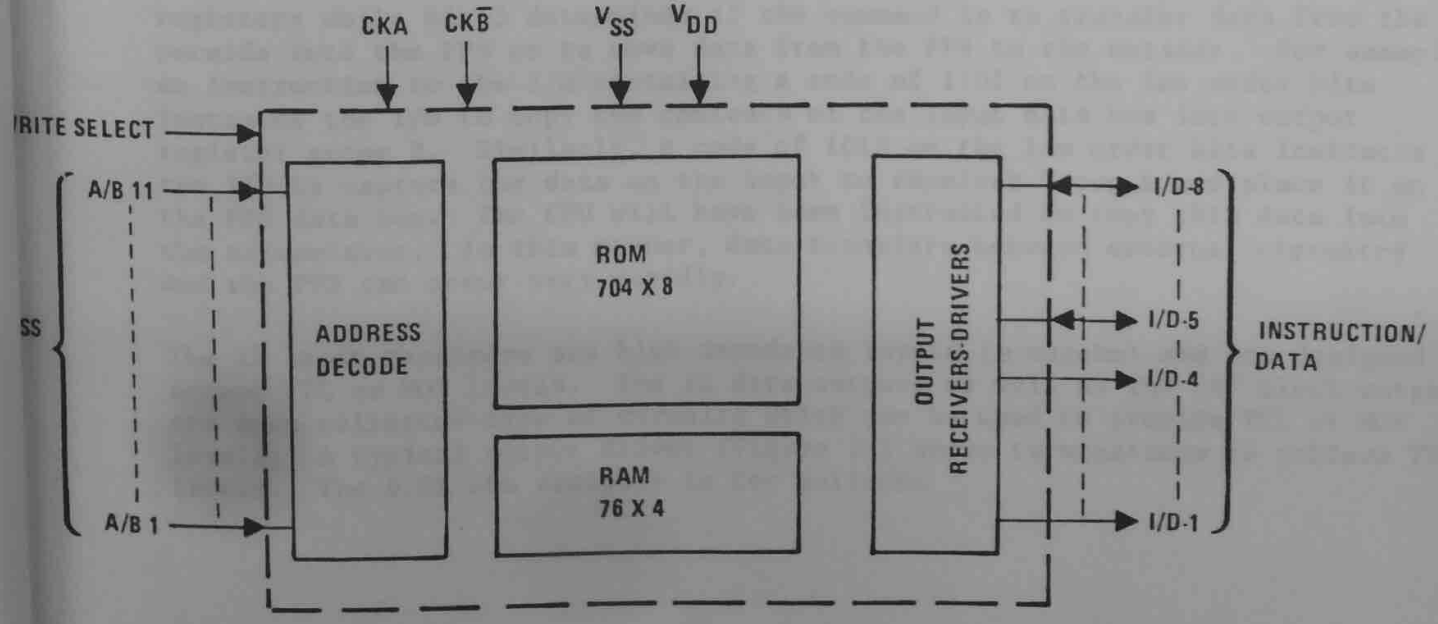


Figure 10. A08--ROM/RAM Block Diagram

E. 10696 General Purpose Input/Output -- The General Purpose Input/Output circuit shown in Figure 11 serves as a timing and buffer interface for exchanging data between the PPS and external equipments. It is designed with dynamic logic for operation from a single 17 volt power supply and is compatible with the PPS. The clock signals, data timing, and impedance loading characteristics of the PPS dual 4-bit parallel data buses are critical for predictable and reliable performance. This performance can only be assumed, when interface to the PPS is through this GP I/O or an I/O designed by NRMEC specifically for this purpose.

The I/O accepts data from and puts data on the data buses as a result of instructions received from the ROM. That is, during one time period, the 8-bit microinstruction on the dual data bus is received by the I/O, if enabled, and decodes 4 bits of this word to determine if it is being addressed and receives the other 4 bits as instructions. The 4-bit instruction word is interpreted by the I/O to either copy the data that follows on the data bus into one of the three 4-bit parallel output registers (A, B, or C) or transfer data from one of the 4-bit parallel input receivers (A, B or C) onto the data bus for copying into the accumulator of the CPU. Table IV lists the commands which the I/O will respond to where groups A, B and C refer to the three groups of 4-bit input receivers and output drivers. The output drivers are static outputs and data remains in the output registers until altered. Bits 1 through 4 of the instruction word are commands to the I/O while bits 5 through 8 are used to address 1 of 16 possible I/O chips. The 4 I/O select inputs are straps which, when terminated by the user, create the addresses for each I/O circuit. The microprogram must be written, of course, to contain the selected address codes. Bits 1, 2 and 4 of the code select one of the three groups of 4-bit registers while bit 3 determines if the command is to transfer data from the outside into the PPS or to move data from the PPS to the outside. For example, an instruction to the I/O containing a code of 1101 on the low order bits instructs the I/O to copy the contents of the input data bus into output register group B. Similarly, a code of 1010 on the low order bits instructs the I/O to capture the data on the input to receiver Group A and place it on the PPS data bus. The CPU will have been instructed to copy this data into the accumulator. In this manner, data transfers between external circuitry and the PPS can occur very rapidly.

The 12 input receivers are high impedance inputs ($\geq$ megohm) and are designed to accept TTL or MOS levels. The 12 data outputs as well as the "A" clock output are open collector type of circuits which can be used to provide TTL or MOS levels. A typical output driver (Figure 12) shows terminations to achieve TTL levels. The 6.8K ohm resistor is for pulldown.

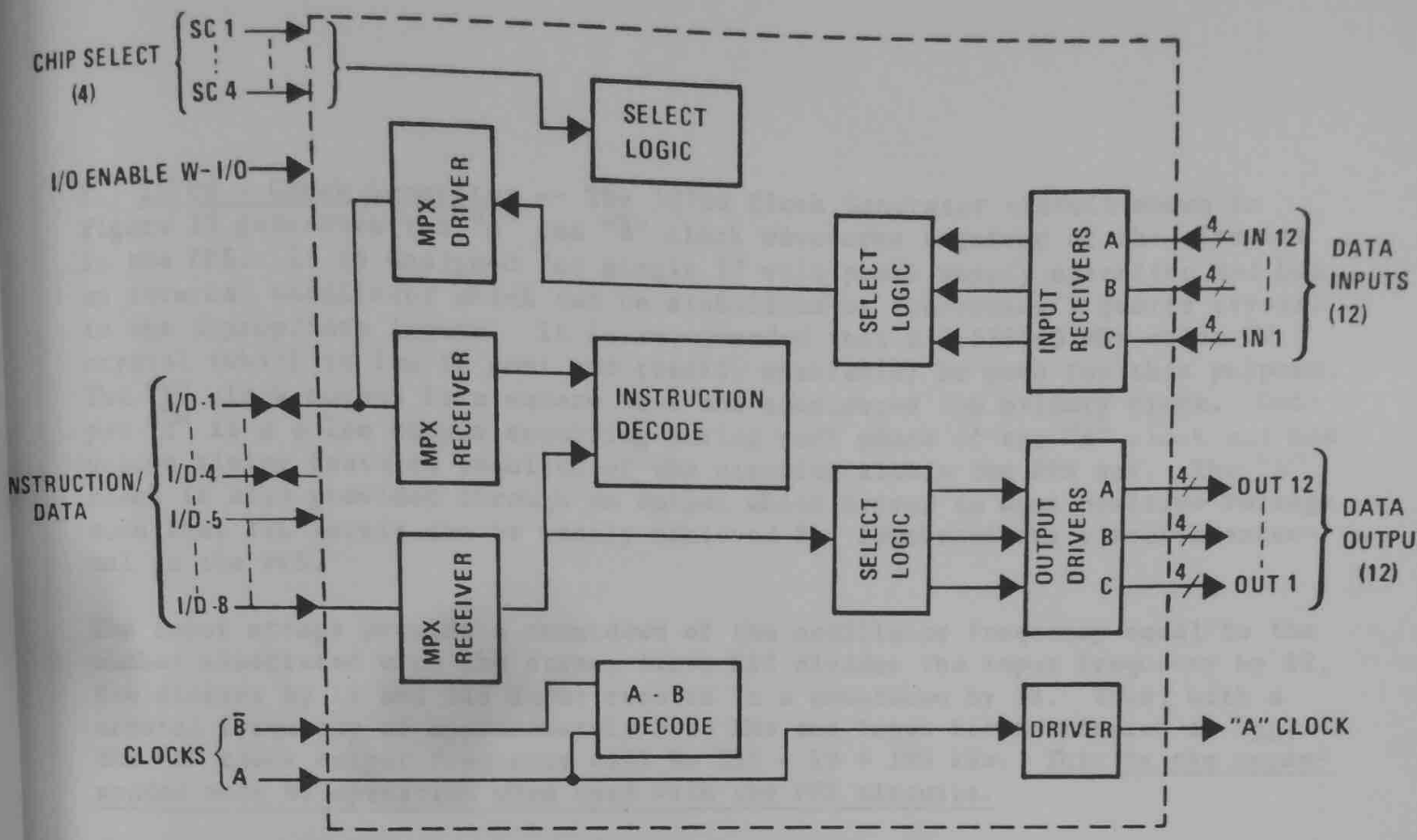


Figure 11. 10696 - General Purpose Input/Output

Table V
DATA BUS I/O INSTRUCTION CODE (I/D)

Chip Select Address	I/D Operation Code								I/D Bus
8	7	6	5	4	3	2	1	
x	x	x	x	1	0	1	0	Read Group A
x	x	x	x	1	0	0	1	Read Group B
x	x	x	x	0	0	1	1	Read Group C
x	x	x	x	1	1	1	0	Set Group A
x	x	x	x	1	1	0	1	Set Group B
x	x	x	x	0	1	1	1	Set Group C

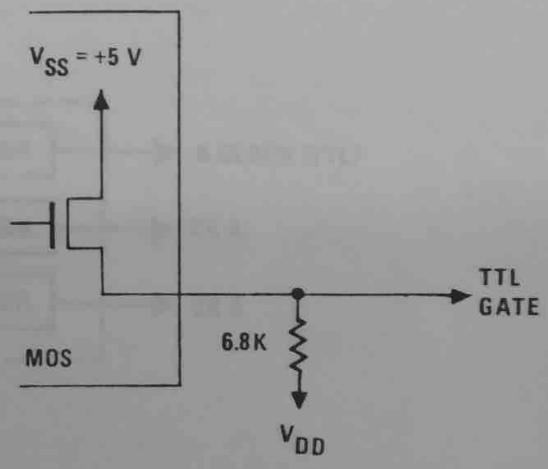


Figure 12 Termination

F. 10706 - Clock Generator -- The 10706 Clock Generator circuit shown in Figure 13 generates the "A" and "$\overline{B}$" clock waveforms required of the circuits in the PPS. It is designed for single 17 volt power supply operation and has an internal oscillator which can be stabilized by connecting a quartz crystal to the appropriate inputs. It is recommended that a 3.579545 MHz color TV crystal (which is low in cost and readily available) be used for this purpose. The "A" clock output is a square wave and considered the primary clock. Output "$\overline{B}$" is a pulse output occurring during each phase of the "A" clock and has unique timing features required of the circuits within the PPS set. The "A" clock is also provided through an output which drives to most positive voltage such that TTL levels can be easily achieved for synchronizing circuits external to the PPS.

The input straps provide a countdown of the oscillator frequency equal to the number associated with the strap; i.e., S12 divides the input frequency by 12, S14 divides by 14 and S18 input results in a countdown by 18. Thus, with a crystal frequency of approximately 3.58 MHz and input S18 terminated to V_{DD}, the "A" clock output frequency will be 358 ÷ 18 = 199 kHz. This is the recommended mode of operation when used with the PPS circuits.

The clock generator can be used to synchronize the PPS to an external system by inputting a square wave to the input of the oscillator instead of connecting a crystal across the oscillator. This allows the user to uniquely determine the operating frequency within the allowed range of the PPS.

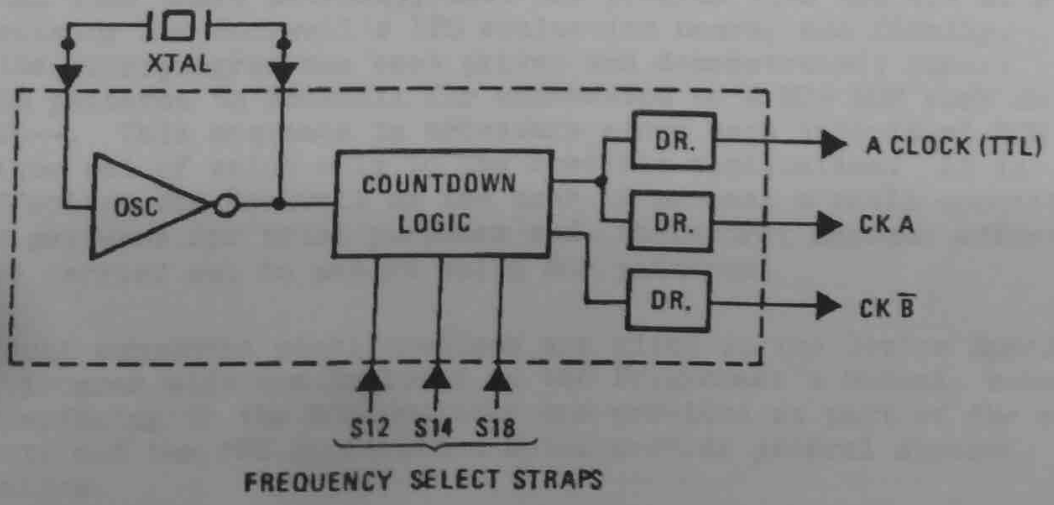


Figure 13. 10706 - Clock Generator Block Diagram

PPS Evaluation Board (20102)

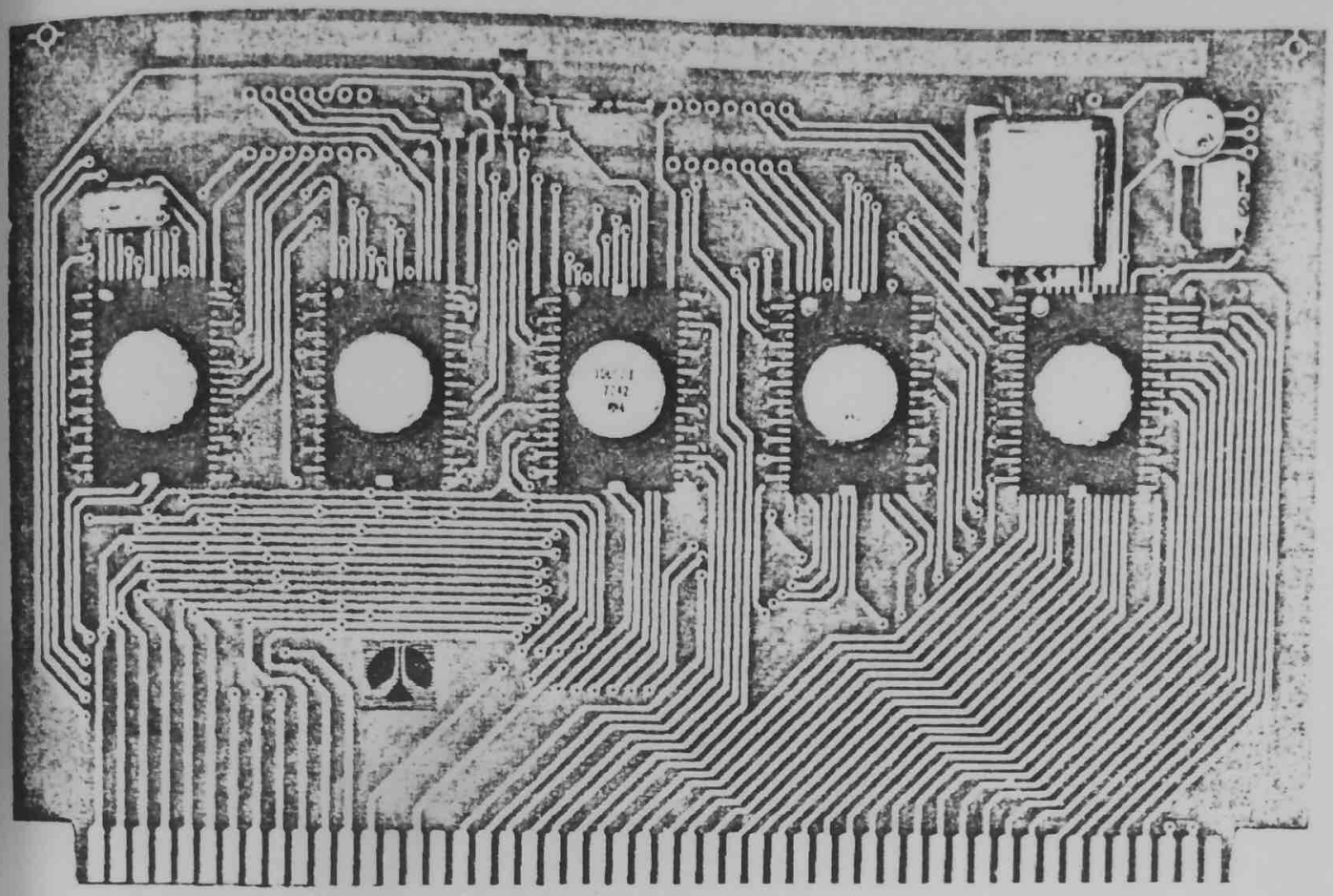
The PPS Evaluation Board is an aid for understanding and utilizing Rockwell's Parallel Processing System. The Evaluation Board is a printed circuit card containing all the necessary components of a PPS set including a crystal controlled clock generator circuit, power-on initialization circuit, one 10660 CPU, two 10432 RAMs, and two 10696 I/Os. That is, it contains everything except the ROM. The content of the ROM is the system microprogram which is developed by the customer and is, therefore, unique to each product and application.

A typical PPS evaluation system is shown in Figure 14. All pertinent signal, control, address, data and I/O lines come off the board. The address and data bus go through a ROM emulator interface circuitry for converting the PPS MOS dynamic signals to TTL static signals. This allows the user to implement the ROM emulator with an in-house minicomputer memory or memory circuits specifically designed to be used for microprogram development such as the Signetics Memory Systems -- SMS Model 1000A ROM Simulator.

The development of a microprogram listing is much like the development of a general purpose computer program. That is, the machine language mnemonics must be understood, the problem must then be defined, an initial program listing compiled, and then -- with the aid of a computer -- the program is tested, debugged and finally released for running.

Thus, a user of PPS must first define the product program problem; compile a listing with the aid of Rockwell's PPS assembler and simulation computer program (available through Tymshare, Inc., national time share network); test the program with the aid of a ROM emulator and Rockwell's PPS evaluation board; and finally, when the microprogram has been proven and demonstrated, submit the ROM patterns to Rockwell for conversion to a MOS ROM such as the A05--. This sequence is necessary since each individual ROM is unique and of value only to the specific application. It is not effective for Rockwell or the user to process a small quantity of ROM patterns for trial purposes and, therefore, maximum effort must be carried out to assure valid ROM patterns.

Individual component specifications are given in the Device Specifications; program aids are included in the Programmer's Manual; schematics for interfacing to the ROM emulator are provided as part of the evaluation kit; and the PPS Application Notes provide general systems information.



PPS Evaluation Board Specifications

1. Board Size -- 5" x 7"

2. System Components on Board

1	10660 CPU
2 ea	10432 RAM
2 ea	10696 General Purpose I/O
1	10706 Clock and Crystal Automatic Power on Reset

3. Power Requirements

17 volts ± 5%
0.060 Amps Typical

4. Board Connector

Sylvania, PN 6AC A0-X7-120-0
Winchester PN HW 50D0-111
Viking PN 3VH50/1CND

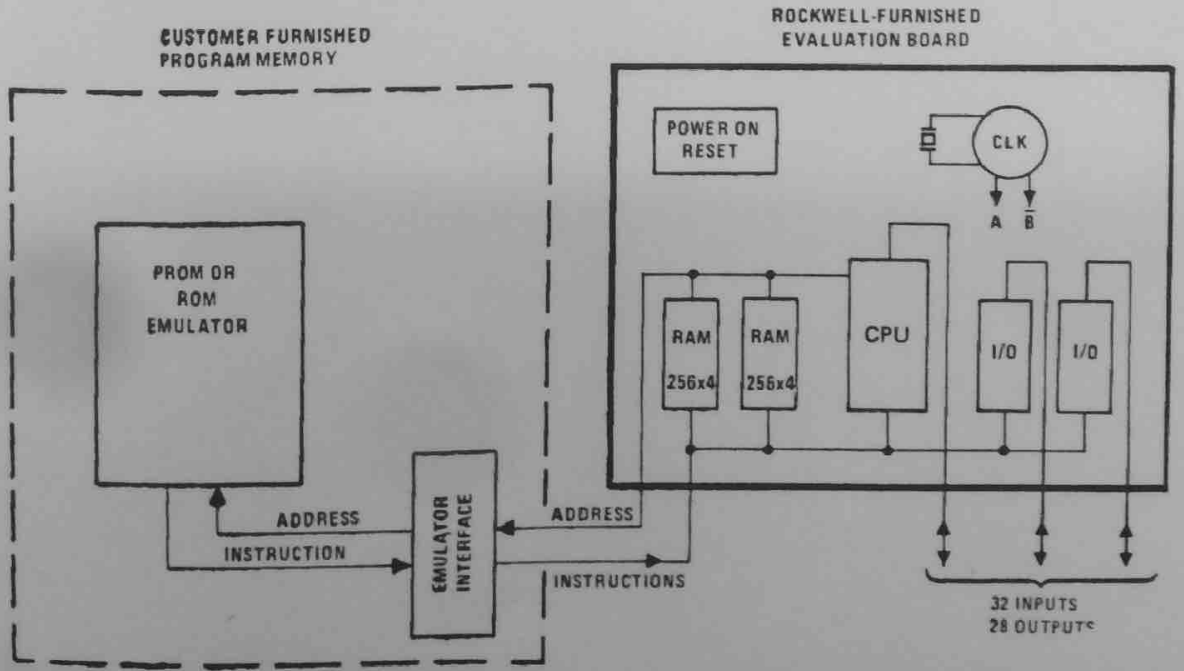


Figure 14 PPS Evaluation System

V

Bean (Whitewall

Hill (Whitewall)

Colglazier (Trucut)

Treleaven (F/O Sales)

DeSwart (Munich)

Smith (Detroit)

Opel (Chicago)

D

PRELIMINARY PRODUCT DESCRIPTION

STC CONTROLLER

(Sequencing - Timing - Counting)

5-25-78

The Rockwell STC (Sequencing, Timing, Counting) controller is the result of a newly implemented concept that enables the power and versatility of microprocessors to be applied to industrial control problems by industrial engineers. No longer is a skilled microcomputer software programmer needed to translate the real world timing, sequencing and counting requirements into the complex language of the computer world. Once a control problem has been reduced to a timing or flow diagram, a sample application of Rockwell's ICOL (Industrial Controller Operat- Language) is all that is needed to program the controller for functional operation.

Rockwell's concept is based on a common microprocessor printed circuit module that has been preprogrammed to operate from commands stored in an insertable PROM. This PROM contains the ICOL coding that customizes the controller for specific applications. Recognizing that there are very few I/O configurations that can be considered standard, it is planned that each application will require a customized I/O module. This module can be provided by yourself using guidelines and application notes from Rockwell or you can have Rockwell or a recommended subcontractor provide it to your specifications.

To further simplify the use of Rockwell STC Controller, it is planned, for the near future, to offer an inexpensive, easy-to-use programmer/checkout system for use in hardware checkout of the desired commands and to program the PROMs needed for operation. The system will accept English ICOL commands and translate them into

the code needed for the PROMs. Specific sets of commands can be stored on or retrieved from audio cassettes and be printed on a paper tape.

The ICOL Commands

One of the important features of the Rockwell Industrial Controller is its ability to be programmed to do its customized functions in a PROM (Programmable Read Only Memory), using a set of commands directly related to the control functions being performed. There will be no need for you to learn all the rules and details of programming a microcomputer, but only the few English type commands created for industrial control type problems. These commands are called ICOL (Industrial Controller Operating Language). Table 1 is a listing of these commands. Table 2 is a listing of the functional elements in the controller. The following is a detailed description of the controller operation and the tables.

The Rockwell Industrial Controller operates sequentially. That is, it starts at the first command, performs it and goes to the next. Each command takes time to be performed. Table 1 gives the specific time for each command.

The Controller's Elements

The term "element" is being used as a generalized descriptor for the functional parts of the controller. These elements consist of discrete type inputs and outputs, digital (4 bit parallel) inputs and outputs, internal timers, an internal time of day (plus day of week) clock, internal counters, status indicators, internal indicators,

TABLE 1

ICOL COMMANDS

Command Name	Command Form	Execution Time (Tentative)	Applicable Elements (f)
IF	IF (f) ON	.8 to 15 ms	indicators, discrete I/O, timers
	IF (f) OFF		" " " "
	IF (f1) = (f2)		timers, display, counters, digital I/O, TOD clock, constants
	IF (f1) < (f2)		ditto above
	IF (f1) > (f2		ditto above
GO TO	GO TO (f)	575 to 595 μs	label
	GO TO (f) & RETURN		"
RETURN	RETURN	170 to 190 μs	--
	VECTOR RETURN		--
TURN	TURN (f) ON	(SKIP) 145 μs	indicators, discrete outputs
	TURN (f) OFF	445 to 465 μs	" " " ,counters, digital outputs
SET	SET (f1) = (f2)	(SKIP) 460 to 470 μs	timers, display, counters, TOD clock, digital output, indicators, discrete outputs
		1 to 4 ms	
TOGGLE	TOGGLE (f)	(SKIP) 145 μs	indicators, discrete output
		500 to 520 μs	
ADD	ADD (f2) to (f1)		timers, displays, counters, TOD clock, digital output
SUBTRACT	SUBTRACT (f2) from (f1)		timers, displays, counters, TOD clock, digital output
SCALE	SCALE (f1) by (f2)		counters, constants, digital outputs
REPEAT	REPEAT (f1) COMMANDS (f2) TIMES	.2 to 1 ms	CONSTANTS
WAIT	WAIT (f) SEC	(SKIP) 470 to 510 ms	constant
	WAIT (f) MIN		"
	WAIT (f) HOURS		"
	WAIT UNTIL (f) IS ON	.5 to 1 ms	timers, discrete input
	WAIT UNTIL (f) IS OFF	(SERVICE) 30 to 150 μs	" " "
CONVERT	CONVERT (f1) to (f2) from (f3)		counters, digital I/O, display, label

TABLE 2

ELEMENTS

ELEMENT TYPE	NO. AVAIL	SPECIFIC REF.NAME	STROBE (IF USED)	DATA SIG.	REMARKS
Digital Inputs	3	D01	DS5-DS8	DID-DIA	Combined for *
		D02	DS1-DS4	DID-DIA	
		D03	DS9-DS11	DID-DIA	only 3 digits
		DIDBL	DS1-DS8	DID-DIA	*for double element
Digital Switch Inputs	1	DSW	--	--	SW S9 - S 21 ON BOARD
Discrete Switch Inputs	8	BSW1	--	--	S1 ON BOARD
		BSW2	--	--	S2 ON BOARD
		BSW3	--	--	S3 ON BOARD
		BSW4	--	--	S4 ON BOARD
		BSW5	--	--	S5 ON BOARD
		BSW6	--	--	S6 ON BOARD
		BSW7	--	--	S7 ON BOARD
		BSW8	--	--	S8 ON BOARD
Discrete Inputs BIS	44	BIS1	SC2	BISA	
		BIS2	SC2	BISB	
		BIS3	SC2	BISC	
		BIS4	SC2	BISD	
		BIS5	SC3	BISA	
		BIS6	SC3	BISB	
		BIS7	SC3	BISC	
		BIS8	SC3	BISD	
		BIS9	SC4	BISA	
		BIS10	SC4	BISB	
		BIS11	SC4	BISC	
		BIS12	SC4	BISD	
		BIS13	SC5	BISA	
		BIS14	SC5	BISB	
		BIS15	SC5	BISC	
		BIS16	SC5	BISD	
		BIS17	SC6	BISA	
		BIS18	SC6	BISB	
		BIS19	SC6	BISC	
		BIS20	SC6	BISD	
		BIS21	SC7	BISA	
		BIS22	SC7	BISB	
		BIS23	SC7	BISC	
		BIS24	SC7	BISD	
		BIS25	SC8	BISA	
		BIS26	SC8	BISB	
		BIS27	SC8	BISC	
		BIS28	SC8	BISD	

(table continued)

Table 2 (continued)
ELEMENTS

ELEMENT TYPE	NO. AVAIL	SPECIFIC REF.NAME	STROBE (IF USED)	DATA SIG.	REMARKS
		BIS29	SC9	BISA	
		BIS30	SC9	BISB	
		BIS31	SC9	BISC	
		BIS32	SC9	BISD	
		BIS33	SC10	BISA	
		BIS34	SC10	BISB	
		BIS35	SC10	BISC	
		BIS36	SC10	BISD	
		BIS37	SC11	BISA	
		BIS38	SC11	BISB	
		BIS39	SC11	BISC	
		BIS40	SC11	BISD	
		BIS41	SC11	BISA	
		BIS42	SC12	BISB	
		BIS43	SC12	BISC	
		BIS44	SC12	BISD	
High Speed Counter Inputs	4	HSC1	SB10	BISA	
		HSC2	SB10	BISB	
		HSC3	SB10	BISC	
		HSC4	SB10	BISD	
Interrupt Vector Controls	4	VECT1	SB11	BISA	
		VECT2	SB11	BISB	
		VECT3	SB11	BISC	
		VECT4	SB11	BISD	
Digital Outputs	3	DO1	SB5-SB8	DOD-DOA	Combined for *
		DO2	SB1-SB4	DOD-DOA	
		DO3	SB9	DOD-DOA	Only 1 digit
		DODBL	SB1-SB8	DOD-DOA	*for double element
Direct Discrete Outputs	2	BOD1	--	--	
		BOD2	--	--	
Discrete Outputs	48	BOS1	SC1	BOSA	
		BOS2	SC1	BOSB	
		BOS3	SC1	BOSC	
		BOS4	SC1	BOSD	
		BOS5	SC2	BOSA	
		BOS6	SC2	BOSB	
		BOS7	SC2	BOSC	
		BOS8	SC2	BOSD	
		BOS9	SC3	BOSA	
		BOS10	SC3	BOSB	
		BOS11	SC3	BOSC	
		BOS12	SC3	BOSD	

(table continued)

ELEMENT TYPE	NO. AVAIL	SPECIFIC REF.NAME	STROBE (IF USED)	DATA SIG.	REMARKS
		BOS13	SC4	BOSA	
		BOS14	SC4	BOSB	
		BOS15	SC4	BOSC	
		BOS16	SC4	BOSD	
		BOS17	SC5	BOSA	
		BOS18	SC5	BOSB	
		BOS19	SC5	BOSC	
		BOS20	SC5	BOSD	
		BOS21	SC6	BOSA	
		BOS22	SC6	BOSB	
		BOS23	SC6	BOSC	
		BOS24	SC6	BOSD	
		BOS25	SC7	BOSA	
		BOS26	SC7	BOSB	
		BOS27	SC7	BOSC	
		BOS28	SC7	BOSD	
		BOS29	SC8	BOSA	
		BOS30	SC8	BOSB	
		BOS31	SC8	BOSC	
		BOS32	SC8	BOSD	
		BOS33	SC9	BOSA	
		BOS34	SC9	BOSB	
		BOS35	SC9	BOSC	
		BOS36	SC9	BOSD	
		BOS37	SC10	BOSA	
		BOS38	SC10	BOSB	
		BOS39	SC10	BOSC	
		BOS40	SC10	BOSD	
		BOS41	SC11	BOSA	
		BOS42	SC11	BOSB	
		BOS43	SC11	BOSC	
		BOS44	SC11	BOSD	
		BOS45	SC12	BOSA	
		BOS46	SC12	BOSB	
		BOS47	SC12	BOSC	
		BOS48	SC12	BOSD	
Display Outputs	8	DSPL	SA4-SA1	A1,B1...H1	combined for *
		DSPL2	SA8-SA5	A1,B1...H1	
		DDSPL1	SA8-SA	A1,B1...H1	*for double ELEMENT
		DSPL3	SA12-SA9	A1,B1...H1	combined for *
		DSPL4	SA16-SA13	A1,B1...H1	
		DDSPL3	SA16-SA9	A1,B1...H1	*for double element
		DSPL5	SA4-SA1	A2,B2...H2	combined for *
		DSPL6	SA8-SA5	A2,B2...H2	
		DDSPL5	SA8-SA1	A2,B2...H2	*for double element

(table continued)

Table 2 (continued)
ELEMENTS

ELEMENT TYPE	NO. AVAIL	SPECIFIC REF.NAME	STROBE (IF USED)	DATA SIG.	REMARKS
		DSPL7	SA12-SA9	A2,B2...H2	combined for *
		DSPL8	SA16-SA13	A2,B2...H2	combined for *
		DDSPL7	SA12-SA9	A2,B2...H2	*for double element
Counters without Battery Backup	16	**CNTR1	--	--	combined for *
		CNTR2	--	--	combined for *
		**DCNTR1	--	--	*for double element
		**CNTR3	--	--	combined for *
		CNTR4	--	--	combined for *
		**DCNTR3	--	--	*for double element
		**CNTR5	--	--	combined for *
		CNTR6	--	--	combined for *
		**DCNTR5	--	--	*for double element
		**CNTR7	--	--	combined for *
		CNTR8	--	--	combined for *
		**DCNTR7	--	--	*for double element
		**CNTR9	--	--	combined for *
		CNTR10	--	--	combined for *
		**DCNTR9	--	--	*for double element
		**CNTR11	--	--	combined for *
		CNTR12	--	--	combined for *
		**DCNTR11	--	--	*for double element
		**CNTR13	--	--	combined for *
		CNTR14	--	--	combined for *
		**DCNTR13	--	--	*for double element
		**CNTR15	--	--	combined for *
		CNTR16	--	--	combined for *
		**DCNTR15	--	--	*for double element
Counters with Battery Backup	16	CNTR17	--	--	combined for *
		CNTR18	--	--	combined for *
		DCNTR17	--	--	*for double element
		CNTR19	--	--	combined for *
		CNTR20	--	--	combined for *
		DCNTR19	--	--	*for double element
		CNTR21	--	--	combined for *
		CNTR22	--	--	combined for *
		DCNTR21	--	--	*for double element

**Available for high speed inputs

(table continued)

Table 2 (continued)
ELEMENTS

ELEMENT TYPE	NO. AVAIL	SPECIFIC REF.NAME	(IF USED)	DATA SIG.	REMARKS
		CNTR23	--	--	} combined for *
		CNTR24	--	--	
		DCNTR23	--	--	*for double element
		CNTR25	--	--	} combined for *
		CNTR26	--	--	
		DCNTR25	--	--	*for double element
		CNTR27	--	--	} combined for *
		CNTR28	--	--	
		DCNTR27	--	--	*for double element
		CNTR29	--	--	} combined for *
		CNTR30	--	--	
		DCNTR29	--	--	*for double element
		CNTR31	--	--	} combined for *
		CNTR32	--	--	
		DCNTR31	--	--	*for double element
Timers	4	TIM1	--	--	
		TIM2	--	--	
		TIM3	--	--	
		TIM4	--	--	
Constants	8	Integer	--	--	Any combination of 2, 4 or 8 digit constants whose total digits don't exceed 16 except only 4 of 2 digits constants are allowed
Flags	33	OVRF	--	--	Overflow flag
		UNDF	--	--	Underflow flag
		VEFLG	--	--	Vector enable flag
		VRFLG	--	--	Vector reset flag
		FLAG1	--	--	
		FLAG2	--	--	
		FLAG3	--	--	
		FLAG4	--	--	
		FLAG5	--	--	
		FLAG6	--	--	
		FLAG7	--	--	
		FLAG8	--	--	
		FLAG9	--	--	
		FLAG10	--	--	
		FLAG11	--	--	
		FLAG12	--	--	
		FLAG13	--	--	
		FLAG14	--	--	
		FLAG15	--	--	

Table 2 (continued)
ELEMENTS

ELEMENT TYPE	NO. AVAIL	SPECIFIC REF.NAME	STROBE (IF USED)	DATA SIG.	REMARKS
		FLAG16	--	--	
		FLAG17	--	--	
		FLAG18	--	--	
		FLAG19	--	--	
		FLAG20	--	--	
		FLAG21	--	--	
		FLAG22	--	--	
		FLAG23	--	--	
		FLAG24	--	--	
		FLAG25	--	--	
		FLAG26	--	--	
		FLAG27	--	--	
		FLAG28	--	--	
		FLAG29	--	--	
Time of Day Clock	1	DAY	--	--	One to Seven days
		CLK	--	---	Hours & Minutes (1 to 24 hours)
		DCLK	--	--	Days, Hours & Minutes (double element)

output signals for eight segment displays, constants, labels and conversion tables. Table 2 is a list of these elements. The elements are to what the commands apply.

ICOL Command Details

In the following description of the commands a double underline will be used to designate the portion of the English input that must be entered, in the same order as it appears below. Single underlines are used to show where the elements are to be entered.

At this point we want to consider the actual way the commands will be entered to the 'Programmer/Checkout' system without getting involved with all the details of operating this previously mentioned support system. Therefore, assume that there will be a typewriter keyboard and that the commands will be entered as if typing a single line of English. Furthermore, assume that all the elements have been given English type (noun) names and that they have been assigned to specific controller I/O or internal functions.

The first word to be typed is expected to be one of the commands. If it is not, it is then assumed to be a label (see following section for description of labels). As in normal English, spaces are used to separate words and number sets. After satisfying all the parts needed for entry of a specific command, comments or notes may be added. A carriage return (CR) ends that line.

The 'IF' Prefix Command

This prefix is used to establish a condition, and if the condition is satisfied, then the command will be performed; otherwise it will be skipped. It's English forms are:

IF (noun) is ON then (command)

IF (noun) is OFF then (command)

The (noun) for the above can be discrete inputs or outputs, status indicators, internal indicators, counters, or time:

IF (noun 1) is EQUAL to (noun 2), then (command)

IF (noun 1) is = (noun 2), then (command)

IF (noun 1) is GREATER THAN (noun 2), then (command)

IF (noun 1) is > (noun 2), then (command)

IF (noun 1) is LESS THAN (noun 2), then (command)

IF (noun 1) is < (noun 2), then (command)

The (noun 1) and (noun 2) for the above can be timers, displays, counters, digital inputs and outputs, time of day clock, indicators or constants.

Note: It has been assumed that neither "to" nor "than" have been assigned as a (noun).
Note: A counter is considered "OFF" if zero and "ON" if not.

The 'GO TO' Command

This command is used to change the normal sequence of commands. It causes the next and subsequent commands to be obtained at a location in the sequence of command identified by a (label). There are two forms of this command, the first is when it is not desired to return to the present sequence of commands and the second is used when it is desired to return to the command following the "GO TO" command. Its English forms are:

GO TO (label).

GO TO (label) and RETURN.

Note: The "GO TO" and "RETURN" command may be used only two times in a sequence without using a "RETURN" command. If more than two 'GO TO and "RETURN" commands

are given, only the last two will be usable for "RETURN" commands.

The "RETURN" Command

This command is used to return the sequence of commands to the command following the last "GO TO" and "RETURN" command or from a vector input. The English forms are:

RETURN
VECTOR RETURN

The "TURN ON" or "OFF" Commands

These commands are used to turn on or off a specific element. Its English forms are:

TURN ON (noun)
TURN OFF (noun)

The noun for this command can be status indicators, internal indicators and discrete outputs.

The "SET" "EQUAL" "TO" Command

This command is used to set one element equal to another. Its English forms are:

SET (noun 1) EQUAL TO (noun 2)
SET (noun 1) = (noun 2)

The (noun 1) and (noun 2) for the above can be timers, displays, indicators, discrete I/O, or counters. In addition (noun 1) can be a digital output and (noun 2) can be time of day clock or a digital input.

Note: It has been assumed that "TO" has not been assigned as a (noun).

The "TOGGLE" Command

This command is used to change an element to on if off and to off if on. Its English form is:

TOGGLE (noun)

The noun for this command can be status indicators, internal indicators and discrete outputs.

The "ADD" Command

This command is used to add two elements together. (Noun 2) is added to (noun 1) and the result is left in (noun 1). Its English form is:

ADD (noun 2) TO (noun 1)

The (noun 1) and (noun 2) for the above can be timers, displays or counters. In addition (noun 1) can be a digital output and (noun 2) can be time of day clock, a digital input or a constant. The constant can be any number from 0 to 9999 for single element and from 0 to 99999999 for double element. If the result is greater than the capacity of the element (i.e., 4 digits for single element or 8 digits for double elements), the overflow flag (OVRF) status indicator will be turned on and will remain on until a subsequent "ADD," "SCALE," or an OVRF "TURN OFF," or "TOGGLE" command is given; then the flag will be set accordingly. The result left in (noun 1) will be accurate to the significant digits allowed. (I.e., if (noun 1) equals 9990 and (noun 2) which equals 15 is added, the resulting (noun 1) would be equal to 0005 and the overflow flag would be on.)

The "SUBTRACT" Command

This command is used to subtract one element from another. Its English form is:

SUBTRACT (Noun 2) FROM (noun 1)

(Noun 2) is subtracted from (noun 1) and this result is left in (noun 1). The (noun 1) and (noun 2) for the above can be timers, displays or counters. In addition (noun 1) can be a digital output and (noun 2) can be time of day clock, a digital input or a constant. The constant can be any number from 0 to 9,999 for a single element and from 0 to 9,999,999 for double elements. If the result is a negative number, the underflow flag (UNDF) status indicator will be turned on and will remain on until a subsequent "SUBTRACT," or an UNDF "TURN OFF" or "TOGGLE" command is given; then the flag will be set accordingly. The result left in (noun 1) will be accurate to the significant digits allowed. (I.e., if (noun 1) equals 0005 and (noun 2) which equals 15 is subtracted, the resulting (noun 1) would be equal to 9990 and the underflow flag would be on.)

The "SCALE" Command

This command is used to scale or multiply an element by a constant. Its English form is:

SCALE (noun) by (constant)

The (noun) can be a counter and the (constant) any number from .01 to 99.99.

The "REPEAT" Command

This command is used when it is desired to repeat a certain sequence of commands for a specified number of times. Its English form is:

REPEAT (constant 1) COMMANDS (constant 2) TIMES

(Constant 1) is the number of commands (1 to 255) to be repeated.

Note: The "GO TO" and "RETURN" command is counted as one command and the commands at the "GO TO" label are not counted.

(Constant 2) is any number from 2 to 16 that represents the number of times that the sequence is to be repeated.

The "WAIT" Command

This command is used when it is desired to stop the progression of commands. Its English forms are:

WAIT (constant 1) SECONDS

WAIT (constant 2) MINUTES

WAIT (constant 2) HOURS

WAIT UNTIL (noun) is ON

WAIT UNTIL (noun) is OFF

Constant 1 is any number from 0.1 to 999.9 seconds in 0.1 second increments or 1 to 9999 seconds in one second increments. Constant 2 is any number from 1 to 9999. The noun for the above is either a timer, a discrete input, or a counter.

Note: It has been assumed that "UNTIL" has not been assigned as a (noun).

Note: A counter at zero is considered off, otherwise it is on.

The "CONVERT" Command.

This command is used to convert the value of a functional element to a new value that is based on a "look up" table. Its English from is:

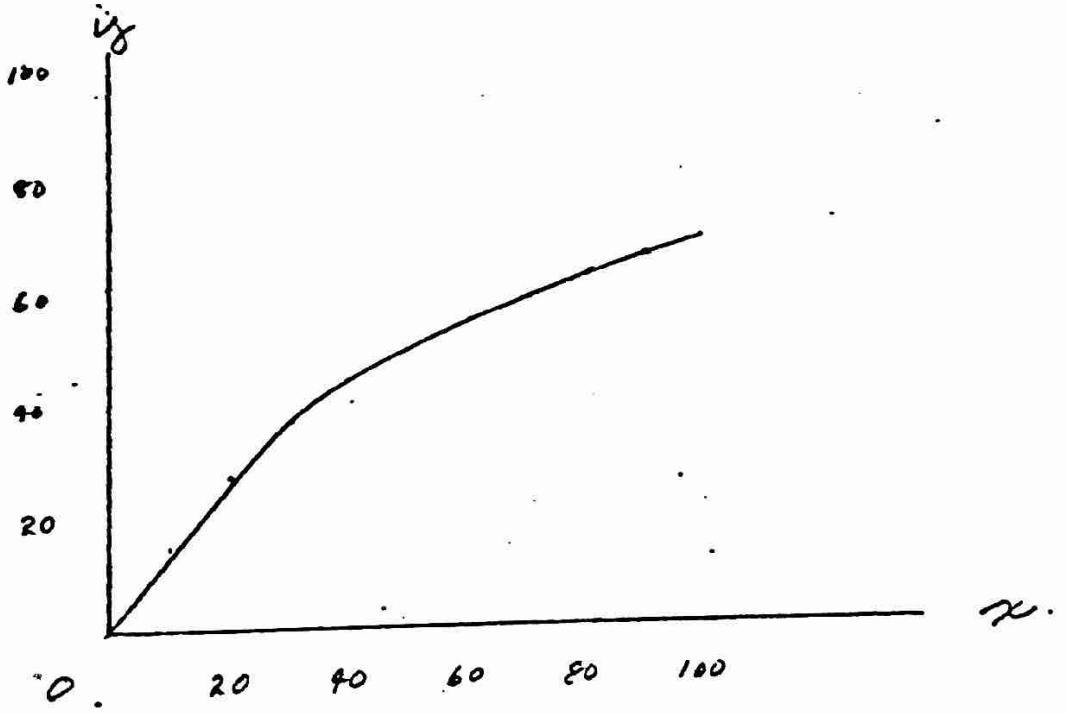
CONVERT (noun 1) TO (noun 2) FROM (label).

(Noun 1) can be a counter or digital input. (Noun 2) can be a counter, a digital output or a display. The (label) indicates the location in the sequence of command where the "look up" table has been placed.

The table consists of eleven user determined values ranging from 00 to 99. If one considers that this value is the "y" ordinate value on a graph and the second least significant digit of (noun 1) is the "x" abscissa value, then this command will result in the second least significant digit of (noun 1) to be used to "look up"

the user determined value at this point and the next, then interpolate between the two points using the least significant digit of (noun 1) as a scale and placing the value in (noun 2).

Graphically, we have



For example, if we let y = 44 at x = 40

y = 50 at x = 50

To find the value of y when x = 46, we have

y when x = 50 - y when y = 40

or 50 -44 = 6

Then interpolating $6 \times \frac{6}{10} = 3.6 = 4$

then $y = 44 + 4 = 48$ @ x = 46

The Controllers Element Details

Table 2 is a summary of the elements used in the STC controller. In order to effectively use the "Programmer/Checkout" system mentioned earlier, a translation must be made from the hardware signals or pin connections to terms recognized by the "Programmer/Checkout" system. Furthermore, it is desired to call these signals with names that have meaning to the person setting up the command sequence. The translation procedure consists of using Table 2 which lists the names that are to be used with the "programmer/Checkout" system. Associated with these names are signal function names that in turn are defined in Table 3 with respect to their actual connector and pin numbers.

For example, say we have a table drive motor that needs to be controlled. Let's choose one of the direct outputs whose name is BODI. From Table 3 we find that this signal is on J7-5 (connector J7 and pin 5). Therefore, we would connect the power driver for this motor to that pin. Now, in setting up the specific command we think of this motor as the table drive motor and a shortened form could be "TBLMTR." So we type in on the "Programmer/Checkout" system keyboard "TBLMTR = BODI.". Later in the sequence of commands if we want to turn the motor on we type "TURN TBLMTR ON" or to turn it off, "TURN TBLMTR OFF." TBLMTR is the (noun) mentioned in previous discussion on the commands.

Element Size

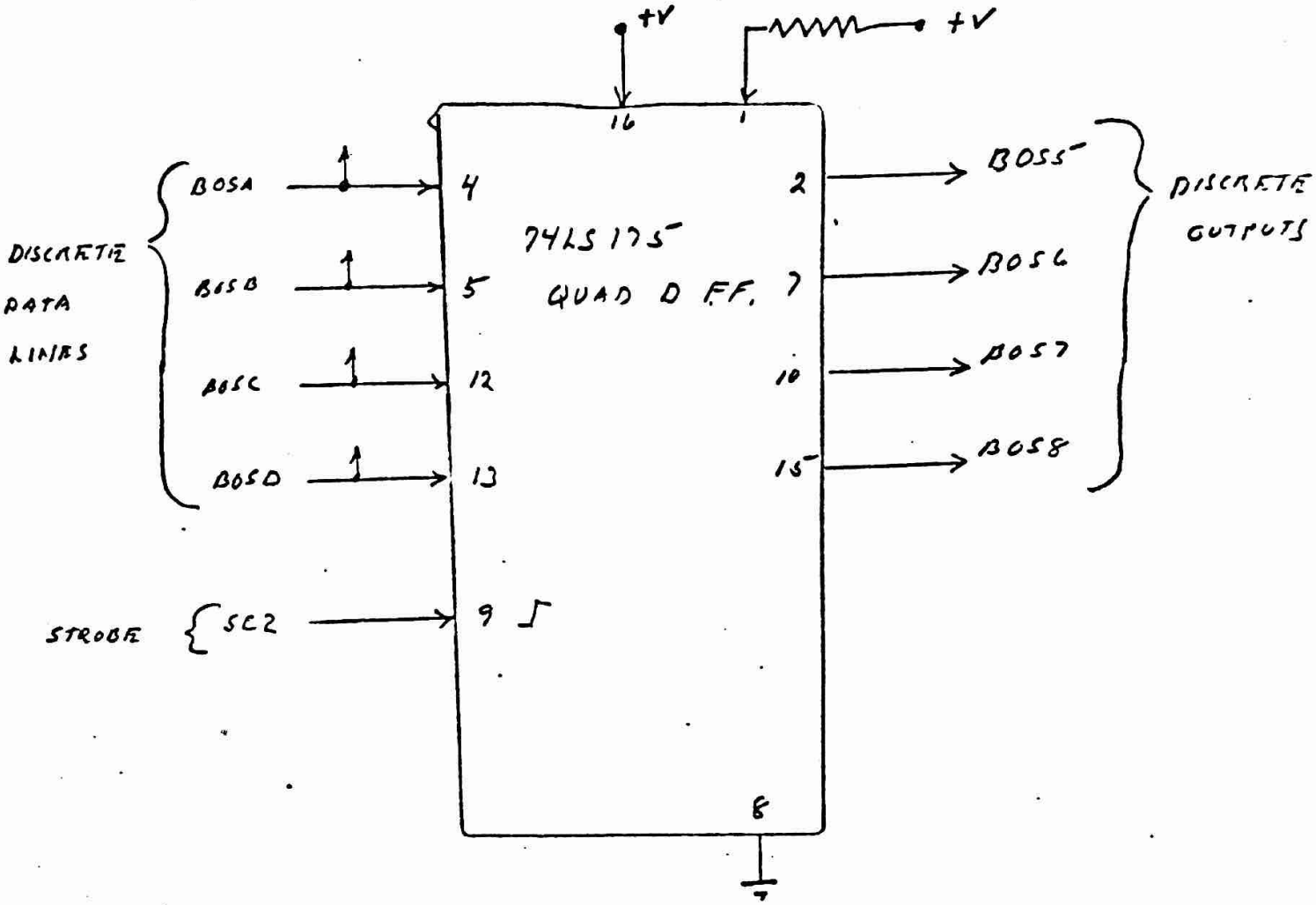
All discrete inputs, outputs and flags are one unit in size and carry the value of "ON" or "OFF." Other elements such as timers, counters, displays and digitals have a basic size of four digits with each digit consisting of four bits. In addition, counters displays and digitals can be paired up to give an eight digit function. A single size element can contain a value of 0 to 9999 with a double size being from 0 to 99999999.

Input/Output Scheme

The STC controller uses a system of strobes with sets of inputs and output lines for most of the input and output functions. There are three sets of strobes: (1) The SA( ) strobe used for the display; (2) the SB ( ) strobe used for digital input and outputs; (3) the SC ( ) strobe used for discrete inputs and outputs. These strobes are active low, which means that a low level signal indicates when that strobe is to be used. In any set of strobes only one line will be low at any one time. Furthermore, the SB ( ) and SC ( ) sets are mutually exclusive, meaning only one of the two sets will be active or low at any time. There are two sets of 8 outputs for the display function, a set of 4 input lines for the digital input, a set of 4 output lines for the digital output, a set of 4 input lines for the discrete inputs,and a set of 4 output lines for the discrete outputs.

For example, four of the discrete outputs could be mechanized by using a 74LS175 Quad D Flip Flop, connecting the outputs BOSA, BOSB, BOSC and BOSD to the four "D" inputs respectively and connecting the SC2 strobe to its clock input. The resulting outputs of

of the EF would be BOS5, BOS6, BOS7 and BOS8, respectively. See Figure.



Inputs

Using Table 2 as a reference, the following is a description of the STC controller's input elements.

Digital Inputs

There are two sets of four digit digital inputs, plus a partial set consisting of three digits. Inputs can be either a BCD type input or a binary type. The STC controller works with BCD type coding but it will accept binary inputs and translate them to the working BCD format. The user must assure that the binary input will not exceed the four digit capacity for BCD coded data or 9999 counts.

Board Switches

There are three eightpole switches mounted on the STC controller board. Two of these switches have been set up to be a digital type BCD or binary input of four digits. The other is set up as a eight discrete inputs.

Discrete Inputs

Another 44 discrete inputs are available as externally multiplexed inputs.

High Speed Counter Inputs

Four more discrete inputs have been designated as high speed counter inputs. If a counter (odd numbered counters 1 - 15 only) is set equal to one of these inputs, the counter will automatically count pulses on these lines for frequencies up to 60 HZ. The pulse width, both on and off portion, must be greater than 7.5 MS. Once set up the counting is continuous unless the counter is recommanded to another task.

Vector Inputs

Four more discrete inputs have been designated as interrupt vector controls. A high level on one of these inputs will cause the STC controller to perform the equivalent of a "GO TO (label) and RETURN" command. When using the "Programmer/Checkout" system the user defines the start of the command to be performed when one of their inputs are received by entering "(label)/VECT(1 to 4)," then the sequence of commands desired and ending with a "VECTOR RETURN " command or by turning the Vector Reset flag ON. The "VECTOR RETURN" command will cause the STC controller to resume control where it was at the time the signal was received. Once one of the Vector signals are received, no more will be acted upon until the "VECTOR RETURN" command is given or the Vector Reset flag is turned on.

Outputs

Digital Outputs

There are two sets of four digit digital outputs plus a partial set consisting of one digit. These outputs can be either a BCD type output or a binary type.

Direct Discrete Outputs

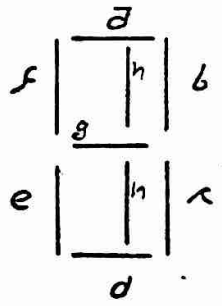
There are two direct discrete outputs that do not require demultiplexing in order to be used.

Discrete Outputs

There are 48 discrete outputs that are mechanized with four data lines and 12 strobes.

Display

There are multiplexed signals for eight four digit displays. These can be paired up to make 8 digit displays. There are 16 strobes to select the digits and two sets of 8 segment signal lines.



Internal Elements

Sixteen of the available 32 four digit counters have battery back up power available. These counters will retain their count even if primary power is removed for periods of several months. Because of the time required to service these battery backed up counters, they can not be set up as high speed counters. However, the eight odd-numbered counters,1 - 15, can. All 32 counters can be selectively paired up to create 8 digit counters.

Timers

There are four, four digit timers available for the user. The value of the digits are controlled by the user. Options are 0.1 sec., sec., min., and hours. When a timer is "SET = TO" the option must also be specified, i.e., "SET TIME EQUAL TO COUNTER SEC." The timer has an error from 0 to 1 unit of the option specified. For example, if one hour is specified, then the timer could run out anywhere from right now up to 1 hour. If 2 hours is specified, then the timer could run out anywhere from 1 to 2 hours. It is therefore important to keep the granularity as small as possible, i.e., if an accurate two hour Wait is desired, specify 7200 sec. The default option on what the STC controller will use if no option is specified is 0.1 sec. per count. Once a timer has been set equal to a value it will start counting down at the option rate specified. At zero it will stop. While the timer is counting down, it is considered ON and when at zero it is considered OFF.

Constants

The STC controller has in addition to its ability for specifying constant values as part of a command, the ability to store up to 16 digits of data in various combinations that can be used for frequently used constants. For example, four of four digit constants can be stored or one of eight digits and four of two digits, etc. These constants are identified with a (noun) name and then referenced by that name in command when they are used. For example, to identify it using the "Programmer/Checkout" system you enter "CONST1 = 24." To use it you enter "Add CONST1 to (Counter 1)" or "IF (counter) is greater than CONST 1." PROM space for commands is saved if this type of constant is used for more than one time.

Flags or Indicators

There are 33 internal flags available for use. Four of them have specified use and are modified by the STC controller's operations. The remaining 29 are fully user controlled and provide means for remembering conditions. For example, if you had four outputs that were in various states and you wanted to turn them all off for a short time, then you wanted to set them back to their original state, the flag could be set the same as the outputs.

I.e., SET FLAGA = OUT 1

Turn the outputs off. Then use the flags to set the outputs.

I.e., SET OUT 1 = FLAGA

The four flags that have specialized use are called: "Overflow Flag" or OVRF, "Underflow Flag" or UNDF, and "Vector Enable Flag" or VEFLG and "Vector Reset Flag" or VRFLG. The overflow flag is set each time an "ADD" or "SCALE" command is given. It is turned on if the results of the command overflows the available digits. If no overflow occurs, then it is turned off. This flag stays in its present state until the next "ADD" or "SCALE" command is given or a command is given to turn it on or off or to toggle it. The underflow flag operates the same as the overflow flag except it is controlled by the "SUBTRACT" command instead of "ADD" or "SCALE" commands and indicates that result was less than zero.

The "Vector Reset Flag" is turned off when one of the four vector inputs turn on. It then prevents any of the other vector inputs from affecting the STC controller operation until this flag is ON. This flag is turned back on when a "Vector RETURN" command is given.

When in the "ON" state, the Vector inputs are again active. This flag is also controlled by the "TURN ON" command. Once it is turned on, the "Vector Return" command is no longer valid until a new "Vector" input is received.

Internal initialization procedures during power on, turns the Vector Enable Flag off, thus disenabling the vector inputs. A user TURN ON VEFLG command must be given to enable the vectors. This flag is also controlled by the "TURN OFF" and "TOGGLE" commands.

e of
ck

The STC controller has a clock option that provides an internal Time of Day (TOD) clock consisting of days (1 to 7), hours (1 to 24) and minutes (0 to 59). If an accurate TTL level 60 HZ 50% duty cycle is applied to the "CLKIN" (J6-2) input the option is available. Setting of the clock is controlled by the multiplexed inputs SET, SEL, and CNT. The SET (Set) places the TOD clock in its set mode. The SEL (Select) is used to select by sequencing which parameter (second, minute, hour or day) is being set. The CNT (control) input causes the selected parameter to slew at a count per second rate. Seconds are set to zero when selected and the "SET" input is turned from on to off.

STC Controller Data

Figure 1 is an outline drawing, showing the physical size and connector location of the STC controller board.

Table 3 is a functional listing of the connections.

Table 4 gives the recommended mating connectors.

Figure 2 gives you typical timing relationships for the multiplexed inputs and outputs.

Table 5 gives the electrical requirements for the board.

The STC controller has been designed to operate from 0°C to 50° C ambient.

I/Φ PIN LIST

ΦUTPUTS

Signal	Pin
Digital Strobes:	
SB1	J5-21
SB2	J5-20
SB3	J5-19
SB4	J5-18
SB5	J5-17
SB6	J5-16
SB7	J5-14
SB8	J5-15
SB9	J5-13
SB10	J4-8
SB11	J4-9
Direct Outputs:	
BΦD1	J7-5
BΦD2	J7-6

Signal	Pin
Discrete Strobes:	
SC1	J5-12
SC2	J5-11
SC3	J5-10
SC4	J5-7
SC5	J5-8
SC6	J5-9
SC7	J5-1
SC8	J5-2
SC9	J4-17
SC10	J4-16
SC11	J4-14
SC12	J4-15

Signal	Pin
Digital Outputs:	
DΦA	J5-6
DΦB	J5-5
DΦC	J5-4
DΦD	J5-3
Discrete Outputs:	
BΦSA	J7-7
BΦSB	J7-4
BΦSC	J7-3
BΦSD	J7-2
Timer Ref:	
TIMREF	?

INPUTS

Digital Inputs:	
DIA	J7-16
DIB	J7-15
DIC	J7-11
DID	J7-10

Discrete Inputs:	
BISA	J7-12
BISB	J7-13
BISC	J7-14
BISD	J7-17

Direct Inputs:	
CLKIN	J6-2
TIMREF	J7-1

TABLE 3

DISPLAY PIN LIST

Signal Name	Connector - Pin
Display Strobes:	
SA1	J3 - 9
SA2	J3 - 8
SA3	J3 - 6
SA4	J3 - 7
SA5	J3 - 17
SA6	J3 - 16
SA7	J3 - 14
SA8	J3 - 15
SA9	J3 - 11
SA10	J3 - 10
SA11	J3 - 12
SA12	J3 - 13
SA13	J3 - 19
SA14	J3 - 18
SA15	J3 - 20
SA16	J3 - 21

Signal Name	Connector - Pin
Display Segments:	
A1	J2 - 3
B1	J2 - 4
C1	J2 - 5
D1	J2 - 6
E1	J2 - 7
F1	J2 - 1
G1	J2 - 2
H1	J2 - 18
A2	J2 - 13
B2	J2 - 14
C2	J2 - 15
D2	J2 - 16
E2	J2 - 17
F2	J2 - 11
G2	J2 - 12
H2	J2 - 19

OTHER

Signal	Pin	
+5V DC	J1-5	Input Power +5V DC
+5V DC	J1-6	" " "
GND	J1-3	Ground (IN)
GND	J1-4	Ground (IN)
-12V DC	J1-2	Input Power -12V DC
BOSD	J6-8	Alternate Output Pins
BOSC	J6-4	" " "
BOSB	J6-5	" " "
BOSA	J6-6	" " "
BISA	J6-3	" " "
BOD2	J6-7	" " "
GND	J3-4	Ground Ref.
GND	J4-1	" "
GND	J5-1	" "
GND	J6-1	" "
GND	J7-8	" "
+5VDC	J3-5	+5V Out
+5VDC	J4-2	+5V Out
+5VDC	J5-23	+5V Out
DOA	J4-10	Alternate Output Pin
DOB	J4-11	" " "
DOC	J4-12	" " "
DOD	J4-13	" " "
Spare	J1-1	Spare Pin
"	J2-8, 9, 10	Spare Pins
"	J3-1, 2, 3	"
"	J4-3, 4, 5, 6. 7	"
"	J5-22	Spare Pin
"	J6-9	"
"	J7-9	"

For Connector	Molex P/N	No. of Circuits	Key Between
J1	09-01-6061	6	2-3
J2	09-01-6191	19	4-5,7-8,11-12,15-16
J3	09-01-6211	21	5-6,10-11,15-16
J4	09-01-6171	17	5-6,10-11,14-15
J5	09-01-6241	24	4-5,8-9,12-13,16-17,20-21
J6	09-01-6091	9	3-4,5-6
J7	09-01-6171	17	5-6,10-11,14-15

Table 4

Mating Connectors

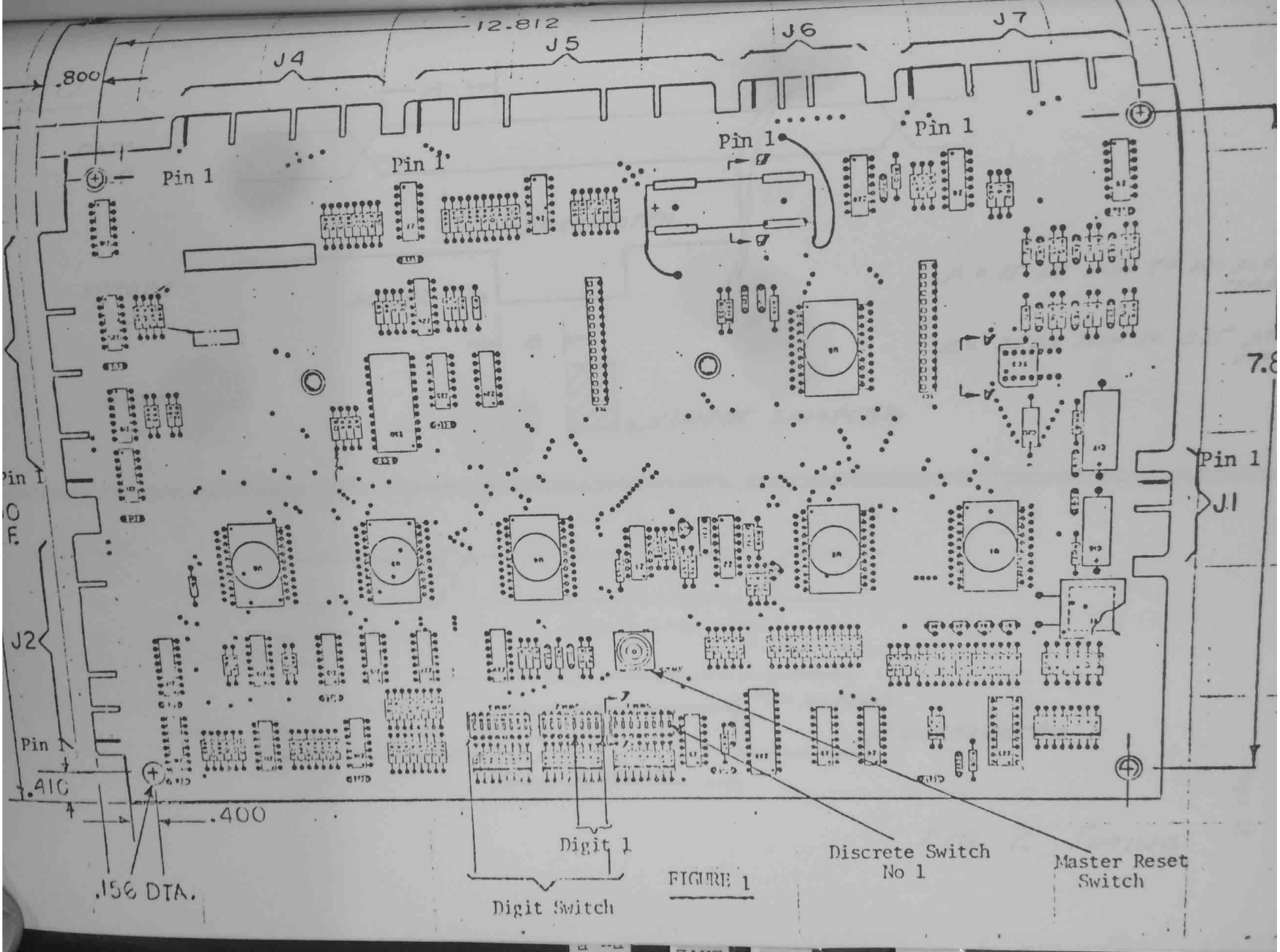

12.812
J4
J5
J6
J7
.800
Pin 1
Pin 1
Pin 1
Pin 1
Pin 1
J1
J2
Pin 1
.410
.400
.156 DIA.
Digit 1
Digit Switch
Discrete Switch
No 1
Master Reset
Switch

FIGURE 1

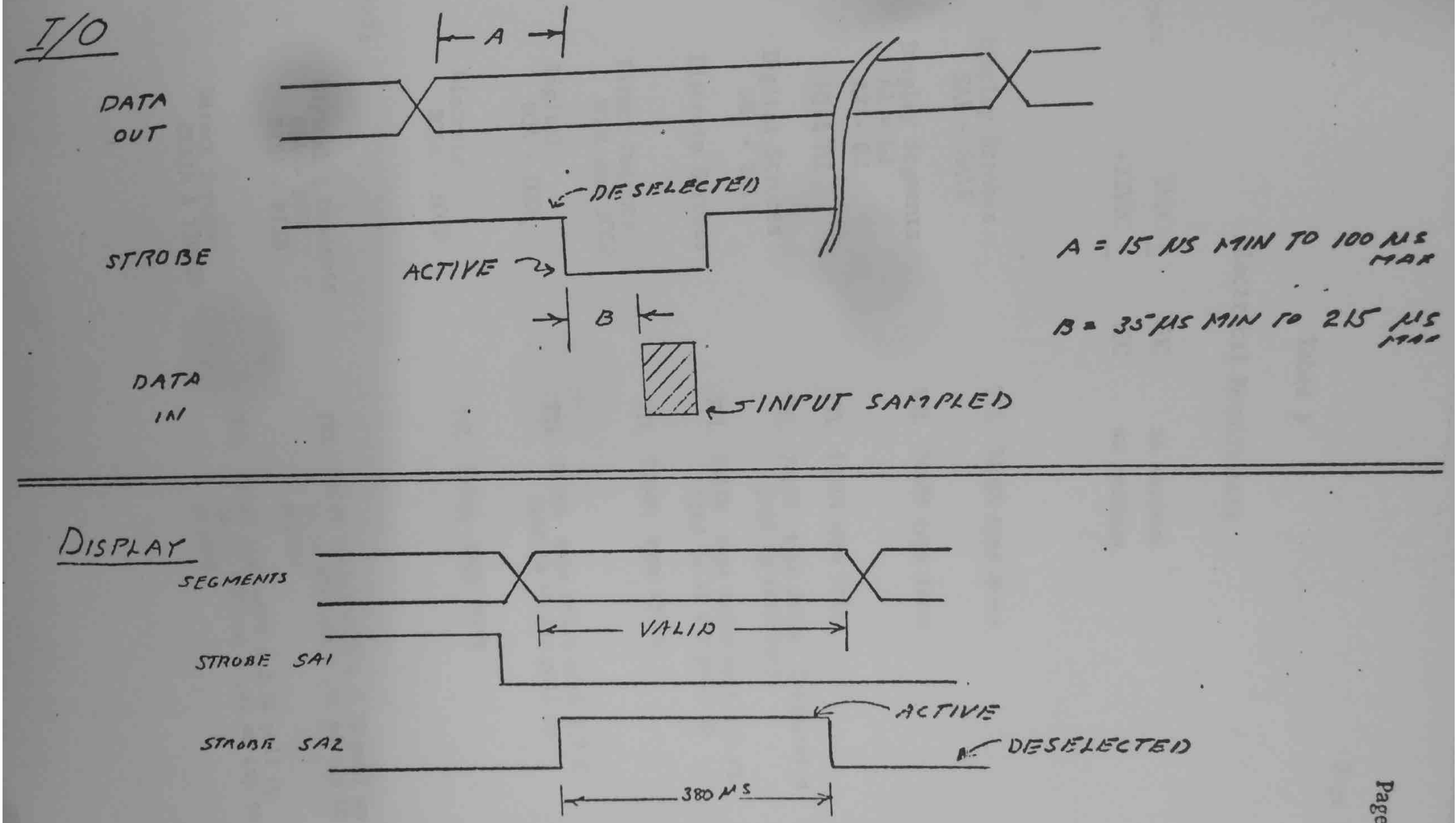

I/O
A
DATA OUT
DESELECTED
STROBE
ACTIVE
B
DATA IN
INPUT SAMPLED
A = 15 MS MIN TO 100 MS MAX
B = 35 MS MIN TO 215 MS MAX
DISPLAY
SEGMENTS
VALID
STROBE SA1
ACTIVE
STROBE SA2
DESELECTED
380 MS

FIG. 2 TIMING

Table 5

Electrical Requirements

Input Power	5VDC ±	VDC	ma maximum
	-12VDC ±	VDC	ma maximum

Outputs

Display Strobes SA1 - SA16	TTL 7408N type drive
Display Segments A1 - G1	TTL 7448N type drive
A2 - G2	TTL 7421N type drive
H1 & H2	TTL 7417N type drive (requires a pull up resistor)
Digital Strobes SB1 - SB11	TTL 7417N type drive with 620Ω tied to +5 for pull up
Discrete Strobes	TTL 7404N type drive
Direct Outputs BOD1 and BOD2	TTL 7416N type drive with 2.7 K tied to +5 for pull up
Digital DOA - DOD	TTL 7404N type drive
Discrete BOSA - BOSD	

Inputs

Digital & Discrete DIA - DID BISA - BISD	TTL 7405N loaded with a 2.7K pull up to +5 VDC and a .01 mf cap to ground
Direct Inputs CLKIN & TIMREF	TTL 7416N load with a 2.7K pull up to +5 VDC and a .01 mf cap to ground

B

new, easy, low cost way to COMPUTERIZE YOUR PRODUCTS

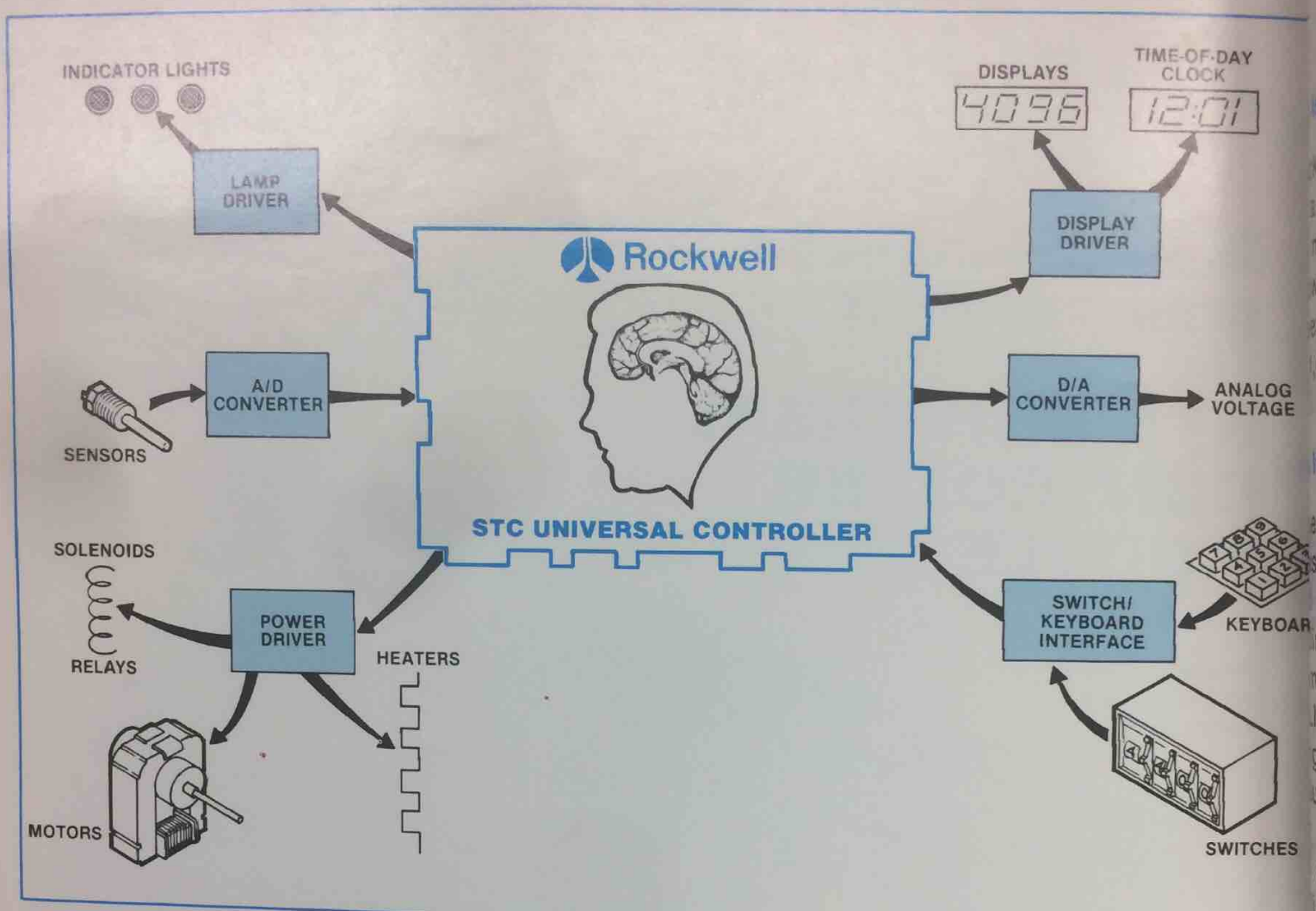
Rockwell

English-language-programmable

STC UNIVERSAL CONTROLLER MODULE

STC Controller — Sequencing, Timing, Counting functions on a pre-designed, pre-tooled module. It keeps your production and design costs low. It's cost effective for many applications . . . in quantities from ONE to 1,000. It's an electronics innovation you should investigate.

The Rockwell STC Universal Controller Module Is Ready-Made for Your Application — And It Understands Plain English Commands . . .



FUNCTIONAL FEATURES

- 12 command types
- 32 four-digit internal counters can be used independently to count to 9,999 or coupled to count to 99,999,999
- 43 inputs, including 4 inputs for high-speed commands and 4 inputs for high-speed counting (up to 60 Hertz)
- 53 outputs
- Digital inputs and outputs can be used in binary or decimal form
- IK × 4 Programmable Read Only Memory (PROM) for user program
- 4 timers provide timing down to 1/10 second intervals
- Signals for 32 digits of display
- On-board battery backup retains the setting of 16 counters, for up to months at a time, when power is turned off or interrupted
- Time-of-Day Clock option allows processes to be monitored and sequenced based on the actual time of day

What It Is:

The Rockwell STC universal controller module is a microprocessor-based controller system mounted on a printed circuit board. Its hardware is pre-designed and pre-tooled. This keeps your price low whether you buy one or 1,000 units.

What's It Good For?

For one piece of equipment or 1,000, you can customize a Rockwell STC module to control operating functions sequentially, according to time or by counting operations or items. This makes it ideal for CONTROLLING pill-filling machines, canning or bottling machines, garment folders, industrial laundry washing machines and dryers, traffic controllers, printing equipment, banks of testing equipment . . . an almost unending list.

How Do You Put It To Work?

There are two design requirements—programming and hardware. And we've simplified these to an elementary level.

Programming: We've developed a unique language of commands by which you develop a program that controls the functions of your customized system. These commands are written in plain English—see back page of this brochure. Even if you know nothing about computers, you're able to write your control program. It's almost as simple as writing, for example: "AT 12:06 AM, TURN MOTOR A ON. WHEN THE EIGHTH ITEM PASSES LEVER C, TURN MOTOR B ON, etc."

Hardware: You do not physically change your STC controller module, but you do have to design interface circuits that "translate" signals to/from your system's components—displays, switches, sensors, motors and the like—into a form that's recognizable by the STC Controller. Rockwell provides you with a variety of applications notes to help you design your interface circuits.

In addition to your interfacing circuitry, which Rockwell can produce for you as a custom module, you need a +5VDC and −12VDC power supply which is commercially available as an off-the-shelf component.

What's Your Next Step?

Send Rockwell a generalized or specific description of the equipment you want to computerize. Include the types of operations you expect to control. The best way to do this is to send a "Ladder" diagram and/or a Timing/Sequence diagram. Please indicate whether you plan to do your own design or whether you want Rockwell to design your complete system.

We will supply you with our recommendations.

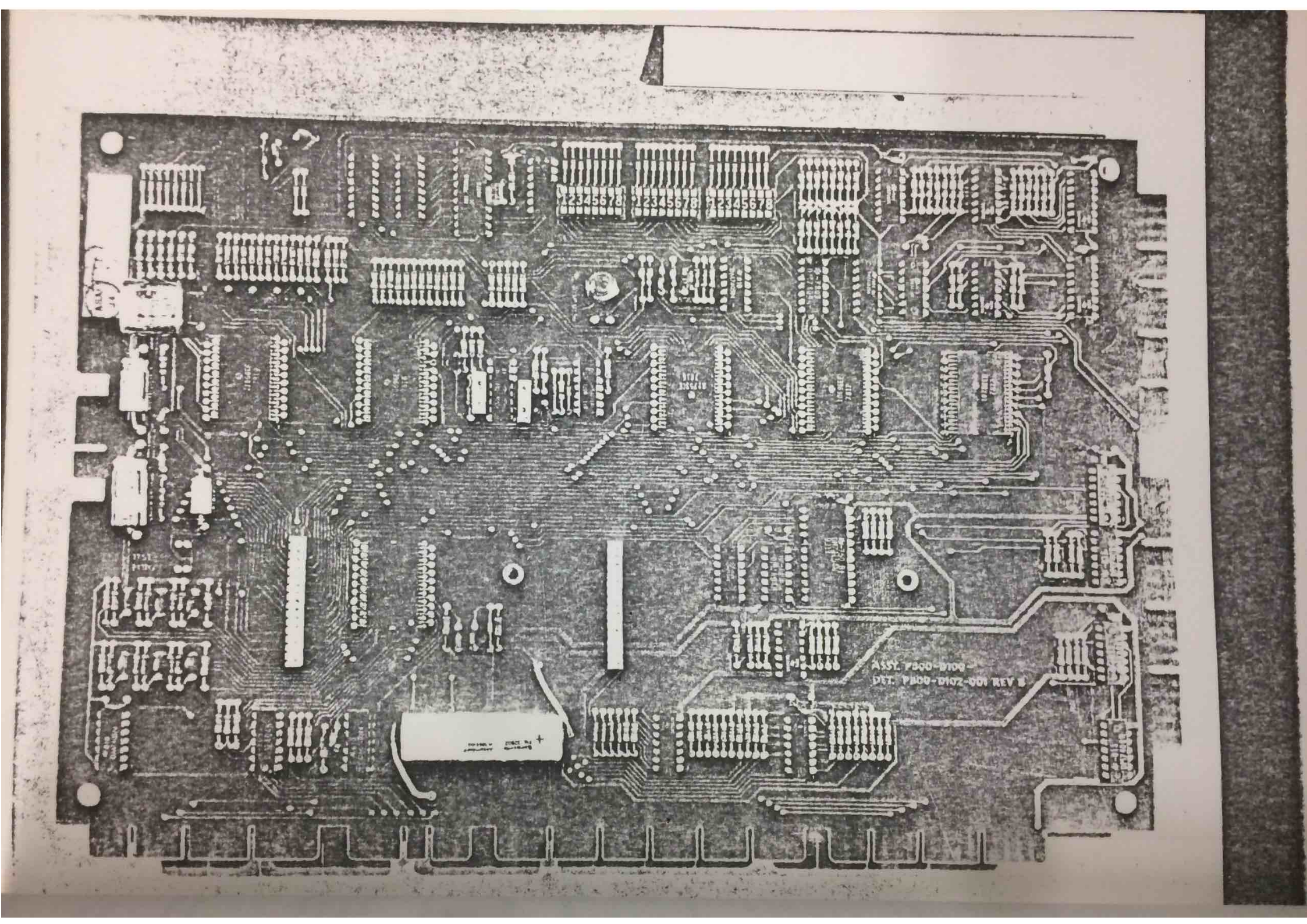
Send your inquiry to:

Subsystem Applications Engineering,
D/833-051 RC33
ROCKWELL INTERNATIONAL
Microelectronic Devices,
P.O. Box 3669,
Anaheim, CA 92803
or phone 714/632-0725.

ROCKWELL STC UNIVERSAL CONTROLLER COMMANDS

COMMAND	FUNCTION(S)	DESCRIPTION
IF *f* IS ON or IF *f* IS OFF	*f* = Status Indicator, Internal Indicator, Discrete Input, Discrete Output, Timer	Causes the next command to be performed only i the specified condition is met.
IF *f1* IS EQUAL TO *f2* or IF *f1* IS GREATER THAN *f2* or IF *f1* IS LESS THAN *f2*	*f1*, *f2* = Timers, Displays, Counters, Digital Inputs, Digital Outputs, Time-of-Day Clock, Constants	Similar to the above IF commands, but operation i conditional upon the relationship of Function *f1* t Function *f2*.
GO TO *label* or GO TO *label* AND RETURN	*label* = Label of any command in program	The next and subsequent commands will be found at the location identified by the specified *label*. A GO TO command is often used after an IF command to tell the program what to do if the specified condi tion is met.
RETURN	None	The next and subsequent commands will be found immediately following the last GO TO *label* AND RETURN command given.
TURN *f* ON or TURN *f* OFF	*f* = Status Indicator, Internal Indicator, Discrete Output	Causes the specified function to be turned on or off.
SET *f1* EQUAL TO *f2*	*f1* = Display, Timer, Counter, Digital Output *f2* = Display Timer, Counter, Digital Output, Digital Input, Time-of-Day Clock, Constant	Causes Function *f1* to be set equal to Function *f2*.
TOGGLE *f*	*f* = Status Indicator, Internal Indicator, Discrete Output	Causes specified Function *f* to change state, turning *f* off if it is on, and vice versa.
ADD *f2* TO *f1* or SUBTRACT *f2* FROM *f1*	*f1* = Display, Timer, Counter, Digital Output *f2* = Display, Counter, Timer, Digital Output, Digital Input, Constant	Causes Function *f1* to receive the result of its addition to or subtraction by Function *f2*. The Overflow Status Indicator is set if an addition yields a sum larger than the "full" condition. The Underflow Status Indicator is set if a subtraction yields a difference that is a negative number.
SCALE *c* BY *n*	*c* = Counter *n* = 00.01 to 99.99	Causes specified Counter *c* to receive the result of its multiplication by *n*. The Overflow Status Indicator is set accordingly.
REPEAT NEXT *n1* COMMANDS *n2* TIMES	*n1* = Number of commands *n2* = Number of repetitions	Causes the next *n1* commands to be repeated *n2* times.
WAIT *s* SECONDS or WAIT *m* MINUTES or WAIT *h* HOURS	*s* = 0.1 to 3600 (Seconds) *m* = 1 to 128 (Minutes) *h* = 1 to 128 (Hours)	Delays the next command for the specified amount of time.
WAIT UNTIL *f* IS ON or WAIT UNTIL *f* IS OFF	*f* = Timer, Discrete Input	Delays the next command until Function *f* has turned on or off, respectively.
CONVERT *f1* TO *f2*, FROM *table*	*f1* = Display, Counter, Digital Output *f2* = Counter, Digital Input *table* = Label of conversion table	Converts Function *f1* to Function *f2*, based on a user-specified conversion *table*. For example, the CONVERT command can be used to convert a non-linear thermistor input into a linearized temperature.

Rockwell International


12345678
12345678
12345678
ASSY. P800-0100
DET. P800-0102-001 REV A

nternal Letter

Rockwell International

No .

ate . July 5, 1978

O (Name, Organization, Internal Address) . J. W. Cox, Jr.
277-022, 031-GA07

FROM: (Name, Organization, Internal Address, Phone) . M. D. LaScala
815, 039-RC48
0624

ubject: . STC Universal Controller Module

ith this IL we are transmitting for your review one (1) STC Universal ontroller Module, P/N PB00-D100. The module is provided under the ollowing conditions:

. The module is loaned to Strategic Systems for 90 days from the date of this IL. It is to be returned to Zane Sandusky.

. Rockwell is currently negotiating a license with D. Gottlieb & Co. for a non-exclusive, world-wide license in and to the data for the purpose of making and selling the STC. The license is for applications other than for or in connection with amusement machines and home electronic games.

. The data and design under license is confidential for all uses. If Strategic Systems incorporates the STC Universal Controller Module in a proposal to any customer including the U. S. Government it shall be accomplished with the proviso that no design information or rights in data will be provided.

. Any contract you negotiate must include a limitation in the data schedule which will exempt the STC design, manufacturing or process data from delivery under ASPR 7-104.9(a) and 7-104.9(b). It would also exclude delivery of data if there is a general predetermination of unlimited rights in data clause requirement.

e cannot express too strongly the sensitivity of the design and applications of the STC Universal Controller Module and we request that you ake every caution internally to maintain its confidential integrity.

. D. LaScala
arketing Administration

PPROVED: C. D. Bopf

Z. Sandusky

c: D. Weber
R. E. McHenry